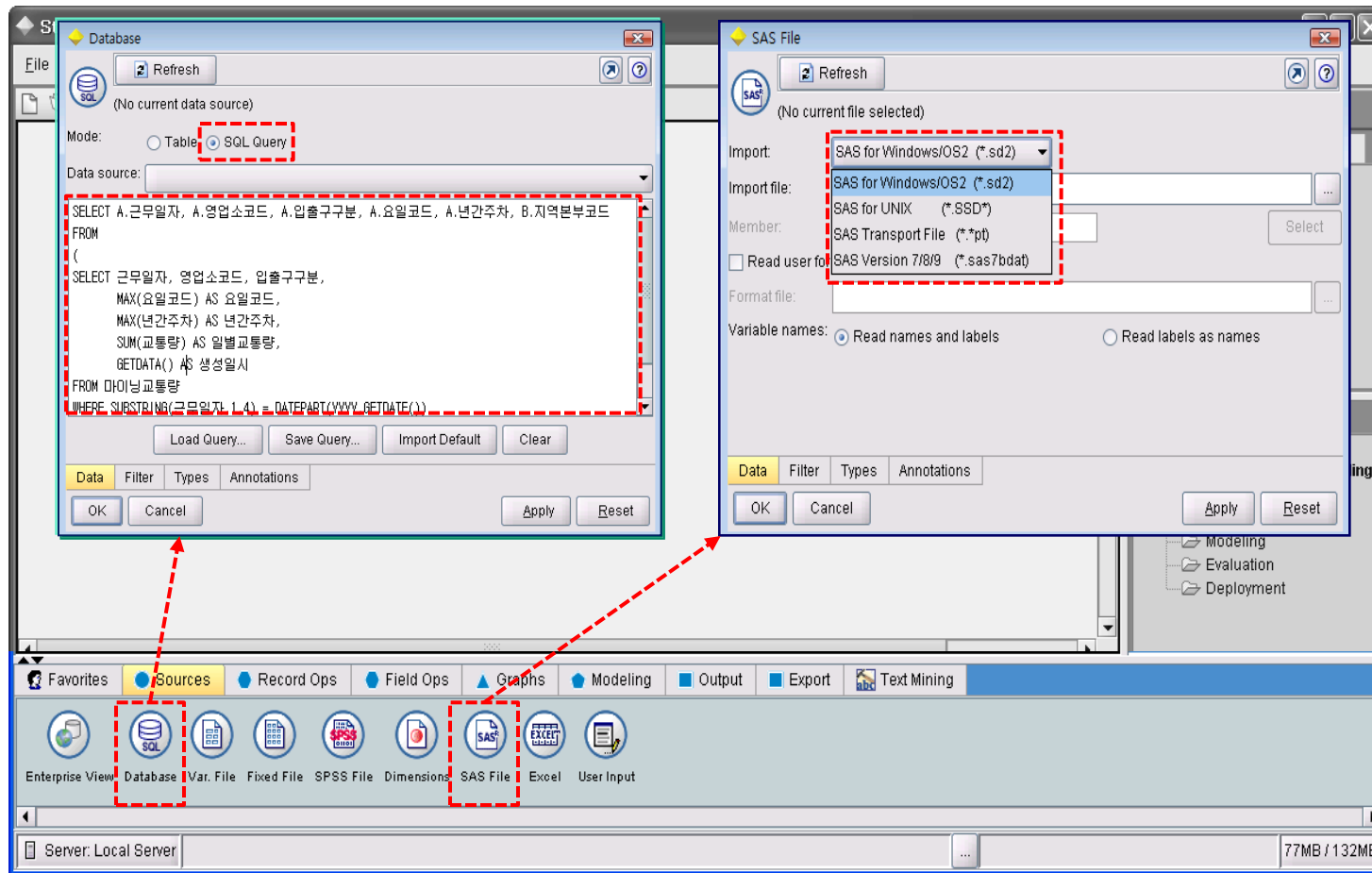




Data Munging with SPSS Modeler

Node의 기능 (1/4)

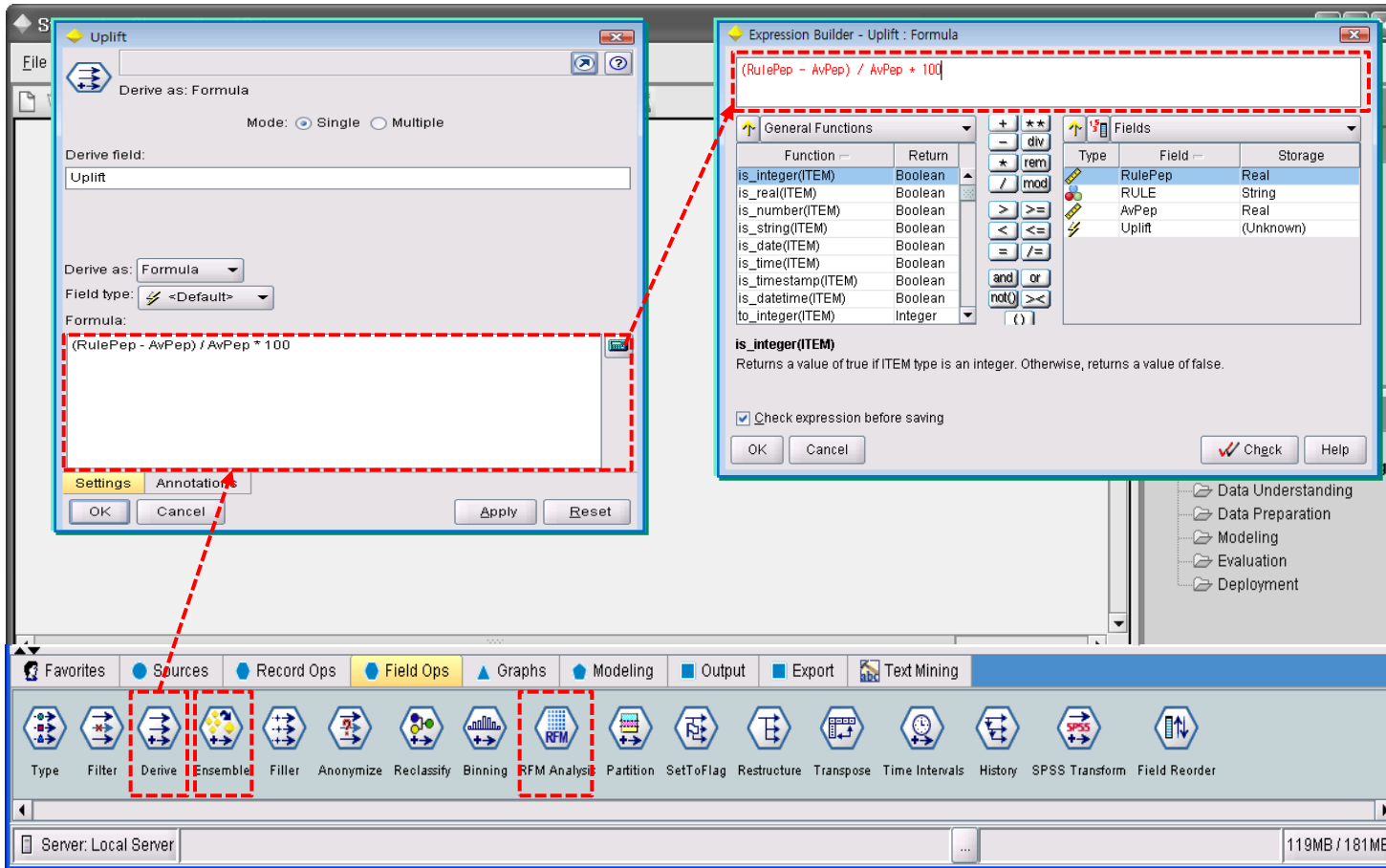


✓ Sources nodes

➤ 분석에 필요한 모든 데이터에 직접 접근 가능 (개방형 구조)

➤ 분석 자료의 저장소(win-win)

Node의 기능 (2/4)



✓ Data handling nodes

➤ 데이터 선택, 생성, 병합 등 다양한 기능들과 Function 으로 Data Manipulation 부분을 유연하게 사용 가능

Node의 기능 (3/4)

의사결정나무 분석

- 최신의 알고리즘 C5.0, Quest 탑재
- 대화식 모형으로 Tree 사용자 정의 가능

연관성 분석

- 다양한 연관성 분석 가능
- 연속형 변수도 가능한 GRI, 동시 후항을 지원하는 CARMA

통계 분석

- 연속형의 선형회귀모형과 범주형의 로지스틱 회귀모형, 요인분석 등 다양한 회귀모형 가능

Screening 모델

- 모델링을 더욱 강력하고, 편리하게 하기 위한 Cleaning 작업
- 군집분석을 이용한 특이치, 이상치 제거 기능

AI기법 / 기타

- 전문가를 위한 신경망의 다양한 Option 조정 가능
- 불완전한 데이터에서 패턴을 찾아내는 Text Mining 기능 (별도)

군집화 / 세그먼트

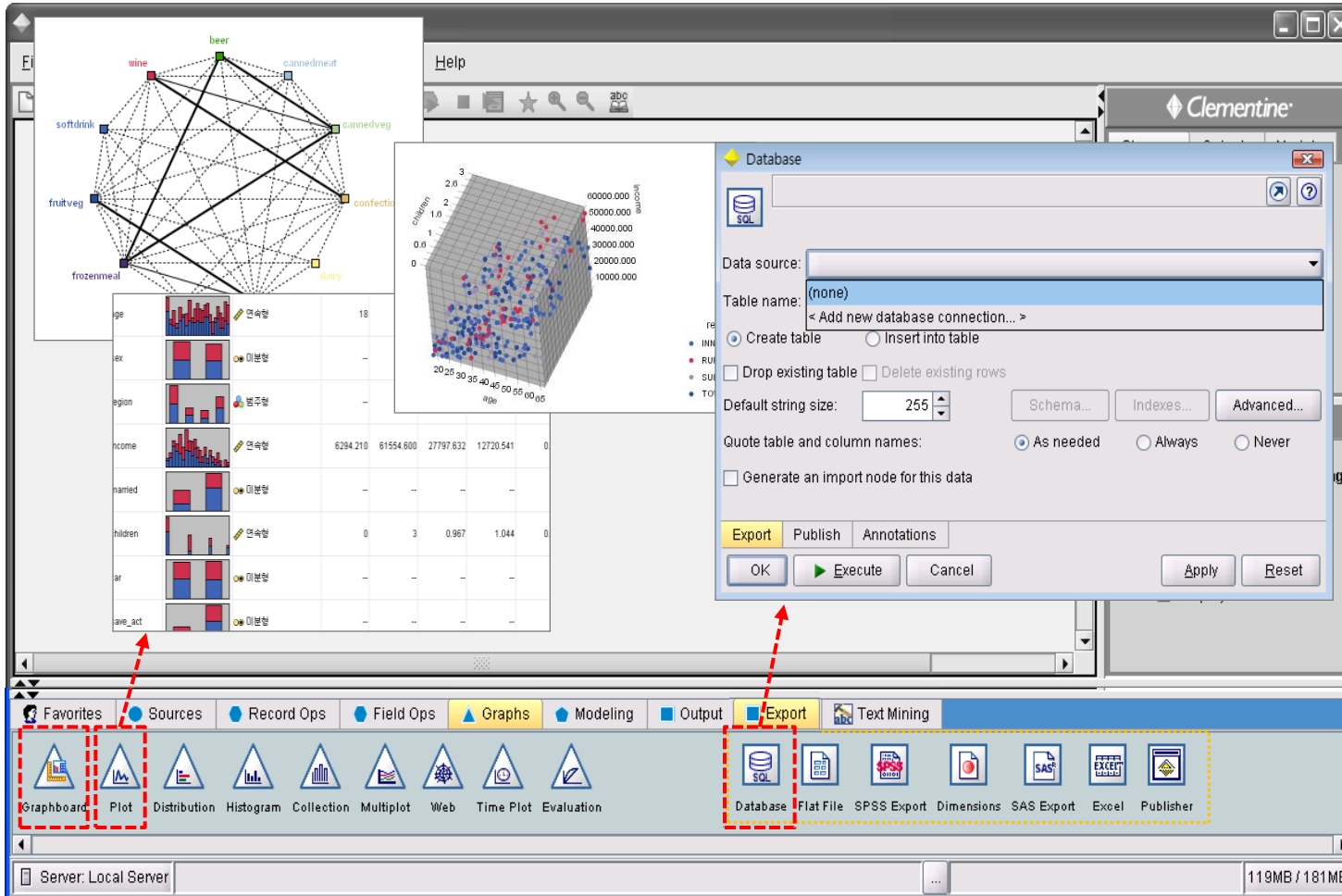
- 대용량 자료의 효율적인 Two-Step Clustering 알고리즘
- 편리한 군집 Rule 찾기 기능

Server: Local Server 119MB / 181MB

✓ Modeling nodes

➤ 가장 많은 종류의 모델링 알고리즘 보유

Node의 기능 (4/4)



✓ Graphs nodes

➤ 다양하고 수려한
그래프와 모델링의
평가 기능

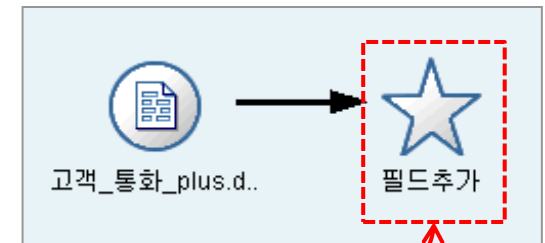
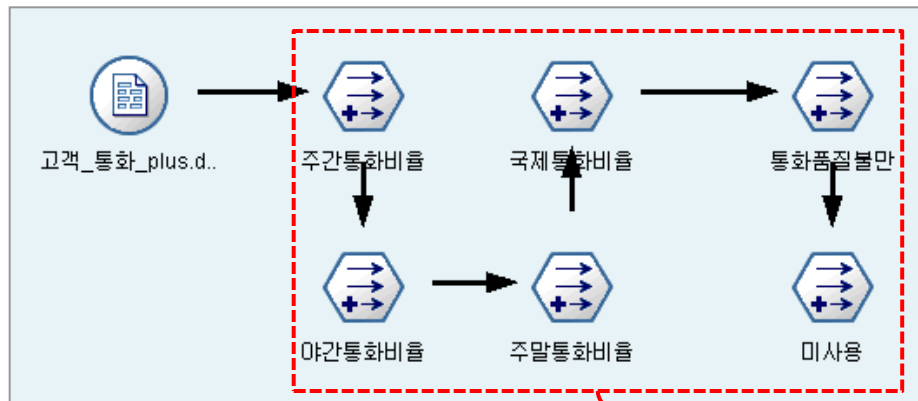
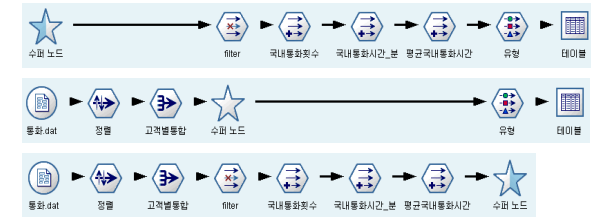
➤ Text, Excel,
SPSS, SAS 파일 등
으로 결과를 출력할
수 있는 기능

Super Node



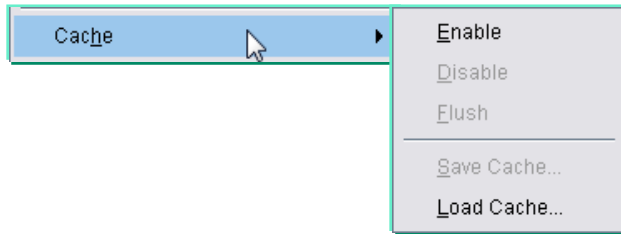
수퍼 노드

- ✓ 여러 노드들을 하나의 노드로 함축 시킴으로 복잡한 노드의 진행을 단순화한다.
- ✓ 복잡한 과정의 반복 수행의 경우 수퍼 노드로 만들어 저장하여 필요 시 불러 사용함으로 작업 시간을 단축 시킨다.
- ✓ 수퍼 노드 내에 다른 수퍼 노드의 사용이 가능하다.

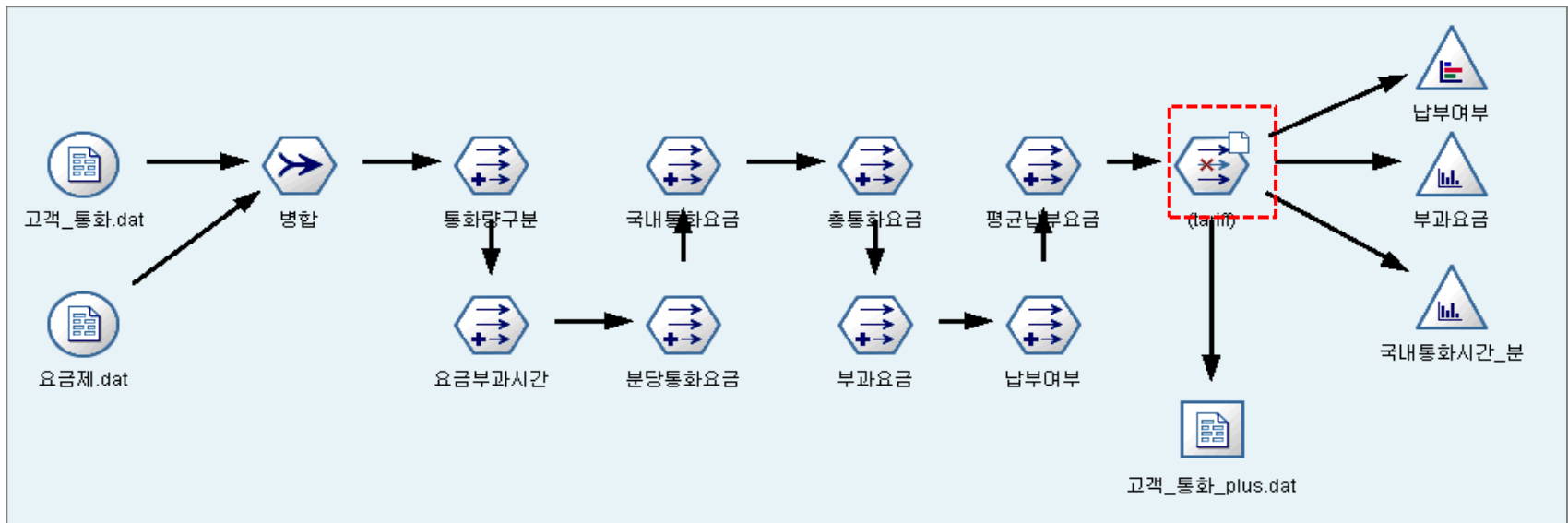


수퍼 노드 사용

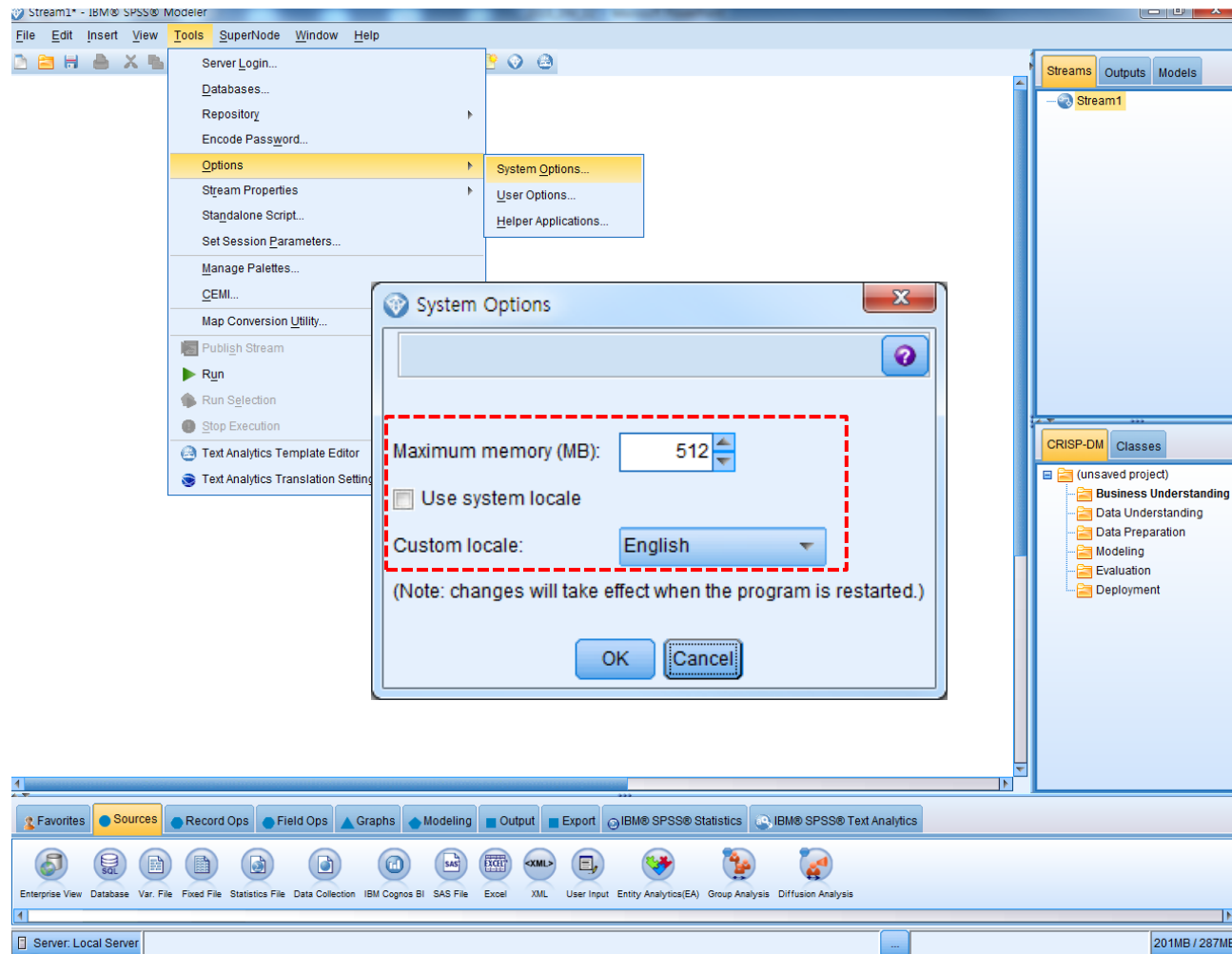
캐쉬(Cashe)



- ✓ 스트림의 특정시점까지의 작업데이터를 일시적 캐쉬에 저장하는 기능
- ✓ 반복적인 탐색의 과정을 수행할 경우에 이전 처리과정을 거친 작업데이터를 캐쉬에 저장하여 사용하면 작업 시간을 단축시킬 수 있다.
- ✓ 캐시로 저장된 작업데이터를 SPSS Save 파일로 저장 또는 Save 파일의 캐쉬 로드가 가능하다.



시스템 옵션 (언어 및 가용 메모리 선택)

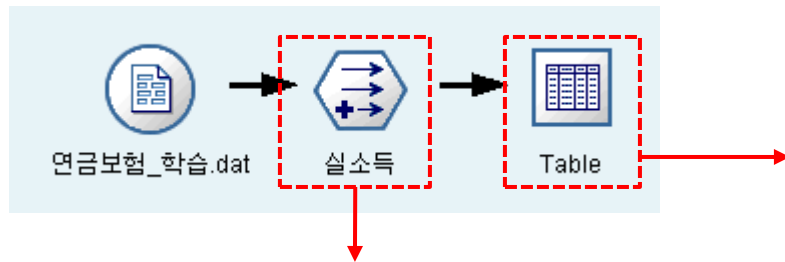


- ✓최대 메모리: Clementine 구동에 활용할 PC의 가용 메모리 설정
- ✓시스템 로케일 사용. 사용자 로케일 선택 가능

변수의 파생

■ 파생(Derive) 노드

: 조건을 부여하여 새로운 필드를 생성한다.



Derive

Derive as: Conditional

Mode: ☒ Single ☐ Multiple

Derive field:

실소득

Derive as: Conditional

Field type: <Default>

If:

자녀수 = 0

Then:

소득

Else:

소득 / (1+0.5*자녀수)

Settings Annotations

OK Cancel Apply Reset

파생 노드로 실
소득을 자녀수를
이용하여 구함

Table (13 fields, 300 records)

	주지역	소득	결혼여부	자녀수	차량소유	저축성	요구불	모기지론	구입여부	실소득
1		17546.000	미혼	1	무	무	무	무	무	11697.333
2		30085.100	기혼	3	무	무	무	무	무	12034.040
3		16575.400	기혼	0	무	무	무	무	무	16575.400
4		20375.400	기혼	3	무	무	무	무	무	8150.160
5		50576.300	기혼	0	무	무	무	무	무	50576.300
6		37869.600	기혼	2	무	무	무	무	무	18934.800
7		8877.070	미혼	0	무	무	무	무	무	8877.070
8		24946.600	기혼	0	무	무	무	무	무	24946.600
9		25304.300	기혼	2	무	무	무	무	무	12652.150
10		24212.100	기혼	2	무	무	무	무	무	12106.050
11		59803.900	기혼	0	무	무	무	무	무	59803.900
12		26658.800	미혼	0	무	무	무	무	무	26658.800
13		15735.800	기혼	1	무	무	무	무	무	10490.533
14		55204.700	기혼	1	무	무	무	무	무	36803.133
15		19474.600	기혼	0	무	무	무	무	무	19474.600
16		22342.100	기혼	0	무	무	무	무	무	22342.100
17		17729.800	기혼	2	무	무	무	무	무	8864.900
18		41016.000	기혼	0	무	무	무	무	무	41016.000
19		26909.200	기혼	0	무	무	무	무	무	26909.200
20		22522.800	기혼	0	무	무	무	무	무	22522.800

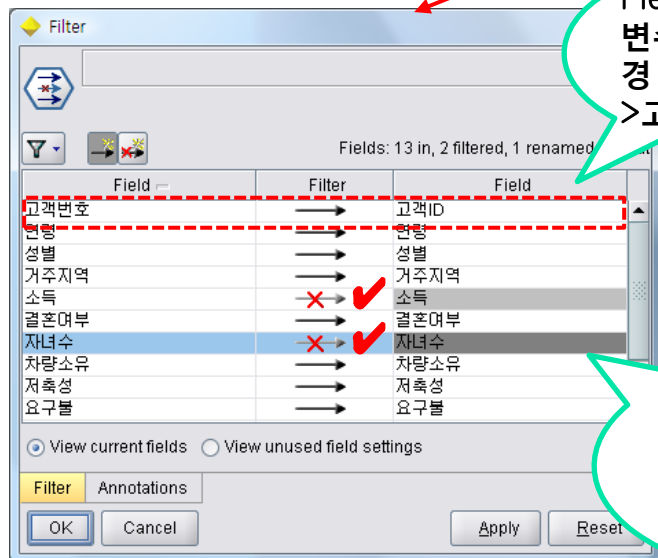
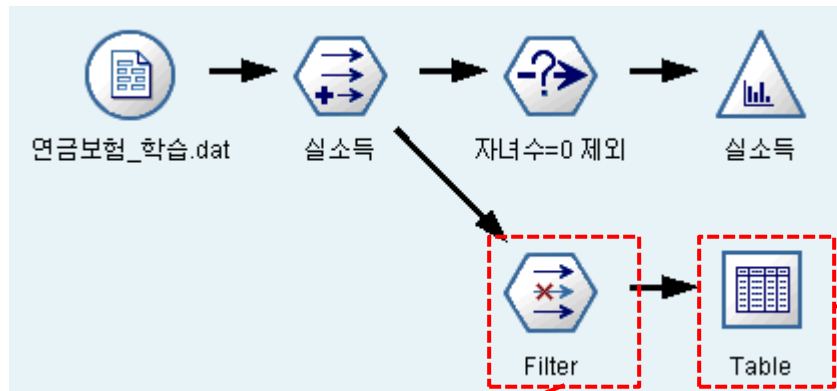
Table Annotations

OK

실소득을 나타내는 새로운 변수를 생성

변수 제거 & 변수명 변경

- 필터(Filter) 노드: 특정한 필드를 선택하거나 제외하고 필드명을 바꾸는 노드



Field 컬럼의 변수이름을 변경 (고객번호 -> 고객ID)

Filter 컬럼의 화살표를 클릭하면 필드의 포함/제거 설정 가능 (소득, 자녀수 제거)

Table (11 fields, 300 records)








	고객ID	연령	성별	거주지역	결혼여부	차량소유	저축성	요구불	모기지론	구매여부	실소득
1	ID12101	48	여자	도시	미혼	무	무	무	무	무	11697.333
2	ID12102	40	남자	도시	기혼	무	무	무	무	무	12034.040
3	ID12103	51	여자	도시	기혼	무	무	무	무	무	16575.400
4	ID12104	23	여자	도시	기혼	무	무	무	무	무	8150.160
5	ID12105	57	여자	시골	기혼	무	무	무	무	무	50576.300
6	ID12106	57	여자	도시	기혼	무	무	무	무	무	18934.800
7	ID12107	22	남자	시골	미혼	무	무	무	무	무	8877.070
8	ID12108	58	남자	도시	기혼	무	무	무	무	무	24946.600
9	ID12109	37	여자	근교	기혼	무	무	무	무	무	12652.150
10	ID12110	54	남자	도시	기혼	무	무	무	무	무	12106.050
11	ID12111	66	여자	도시	기혼	무	무	무	무	무	59803.900
12	ID12112	52	여자	도시	미혼	무	무	무	무	무	26658.800
13	ID12113	44	여자	도시	기혼	무	무	무	무	무	10490.533
14	ID12114	66	여자	도시	기혼	무	무	무	무	무	36803.133
15	ID12115	36	남자	시골	기혼	무	무	무	무	무	19474.600
16	ID12116	38	여자	도시	기혼	무	무	무	무	무	22342.100
17	ID12117	37	여자	도시	기혼	무	무	무	무	무	8864.900
18	ID12118	46	여자	근교	기혼	무	무	무	무	무	41016.000
19	ID12119	62	여자	도시	기혼	무	무	무	무	무	26909.200
20	ID12120	31	남자	도시	기혼	무	무	무	무	무	22522.800
21	ID12121	61	남자	도시	기혼	무	무	무	무	무	28940.350

테이블 노드로 변경사항 확인

변수 유형 설정 (1/2)

변수의 유형과 역할

필드 유형

Icon	Measurement level
	Default
	Continuous
	Categorical
	Flag
	Nominal
	Ordinal
	Typeless

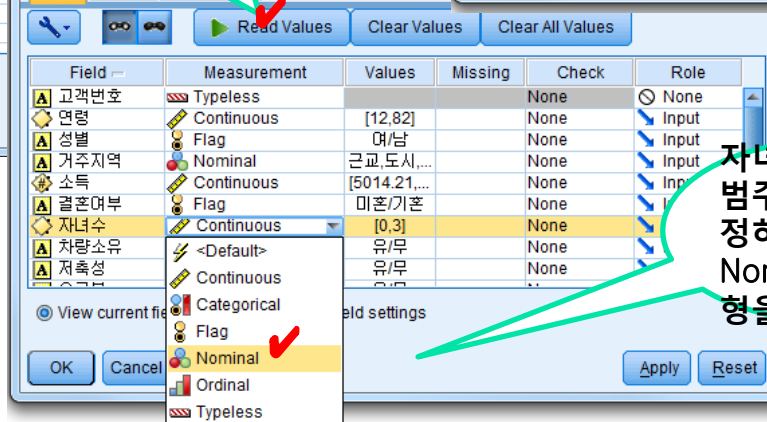
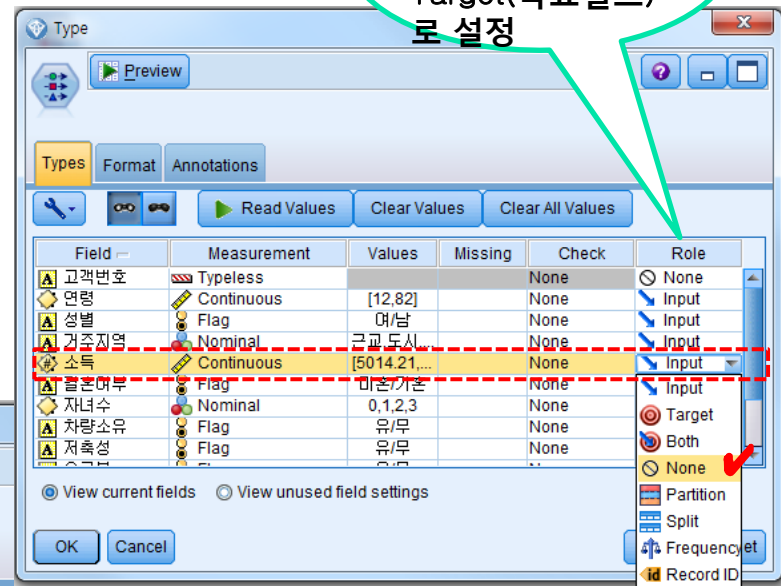
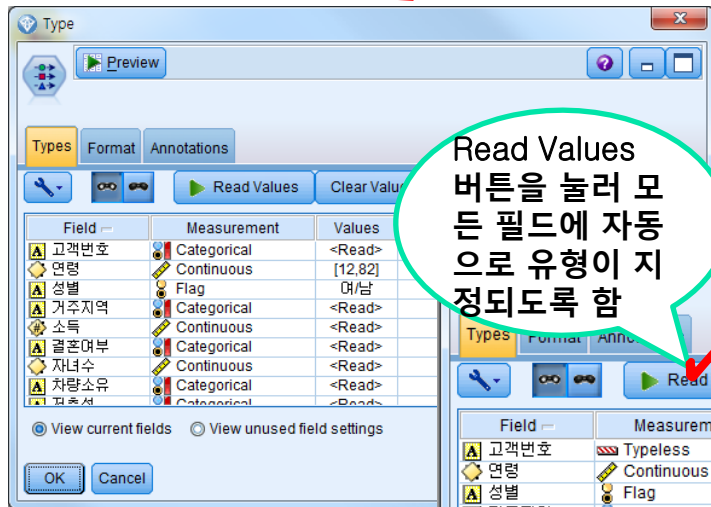
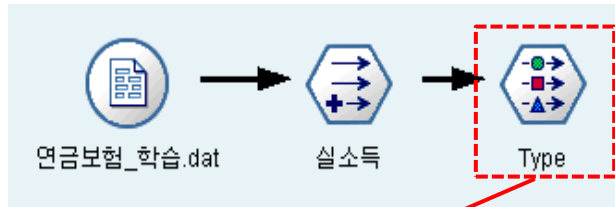
필드 역할

	Input	입력필드
	Target	출력필드
	Both	양방향필드
	None	사용안함

변수 유형 설정 (2/2)

■ 유형(Type) 노드

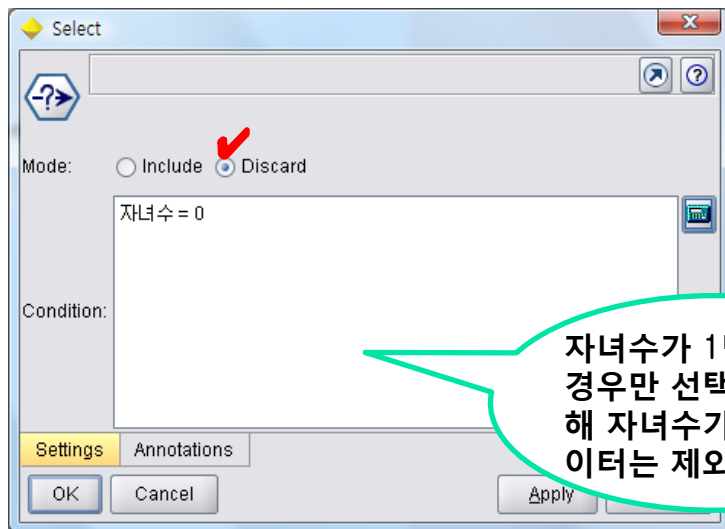
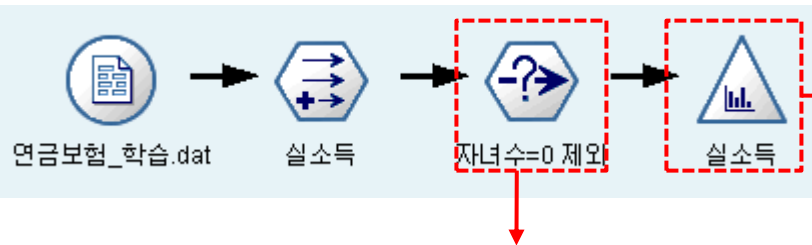
: 필드의 유형(type), 누락값(missing value), 방향(direction)을 설정한다.



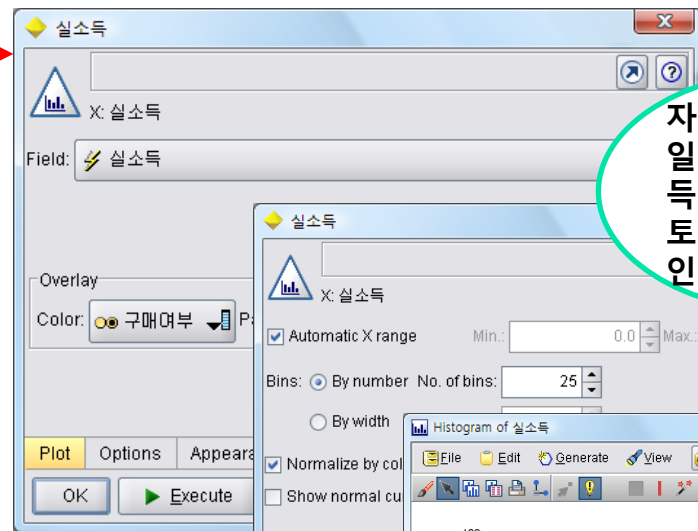
Data의 선택

■ 선택(Select) 노드

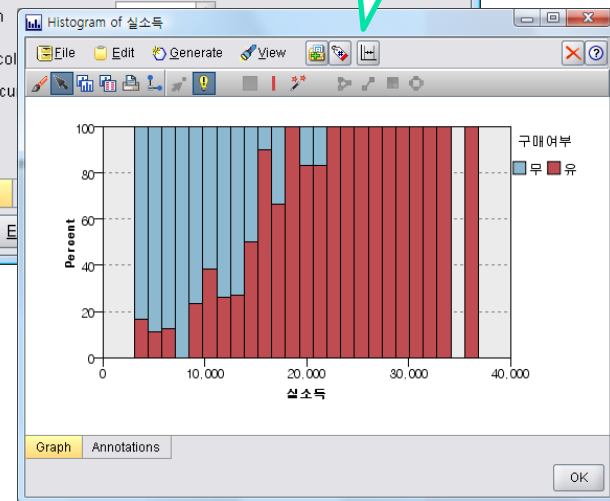
: 일정 조건을 만족하는 특정 레코드를 선택이나 제외하는 노드이다. 조건에는 선택이나 제외에 대한 조건을 지정한다.



자녀수가 1명 이상인
경우만 선택하기 위
해 자녀수가 0인 데
이터는 제외함



자녀수가 1명 이상
일 때 생성된 실소
득의 분포를 히스
토그램을 통해 확
인





Data의 결합

- 무엇을 하는가?
 - File과 file, data와 data를 결합한다.
- 왜 하는가?
 - Data들이 절대로 한 file에 있지 않다.
 - 결합을 통한 DATA의 표준/통합 작업
 - 가장 먼저 직면하는 Manipulation 작업
- 어디에서?
 - 거의 모든 Data Mining Process에서 사용된다.

사용되는 Node



■ 추가(Append) 노드

- 서로 같은 변수끼리의 결합, 레코드의 증가
- 종적 결합에 이용



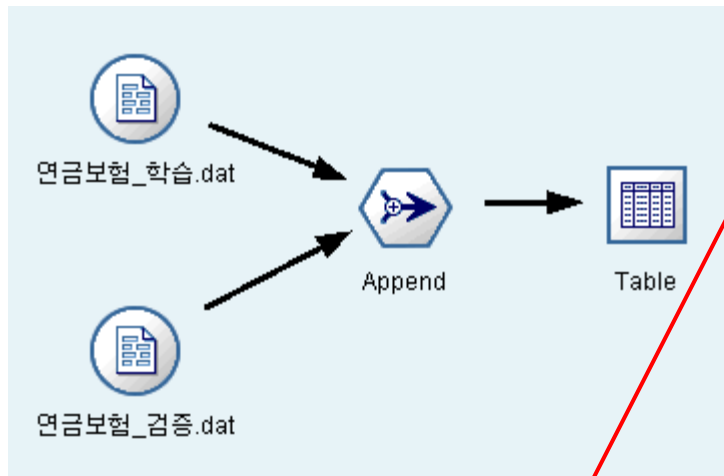
■ 병합(Merge) 노드

- 서로 다른 변수끼리의 결합, 변수의 증가
- 횡적 결합에 이용

Append Node (1/2)

실습 DATA

- 연금보험_학습.dat
- 연금보험_검증.dat



- 연금보험_학습 (12개 변수, 300개 데이터)

This screenshot shows a data table window titled 'Table (12 fields, 300 records) #1'. The table contains 12 columns: 고객번호, 연령, 성별, 거주지역, 소득, 결혼여부, 자녀수, 차량소유, 저축성, 요구불, 모기지론, and 구매여부. The first 8 rows of data are visible, showing various customer profiles.

	고객번호	연령	성별	거주지역	소득	결혼여부	자녀수	차량소유	저축성	요구불	모기지론	구매여부
1	ID12101	48	여자	도심	17548...	미혼	1	무	무	무	무	무
2	ID12102	40	남자	도시	30085...	기혼	3	유	무	무	무	무
3	ID12103	51	여자	도심	16575...	기혼	0	유	무	무	무	무
4	ID12104	23	여자	도시	20375...	기혼	3	무	무	무	무	무
5	ID12105	57	여자	시골	50576...	기혼	0	무	무	무	무	무
6	ID12106	57	여자	도시	37869...	기혼	2	무	무	무	무	무
7	ID12107	22	남자	시골	8877.0...	미혼	0	무	무	무	무	무
8	ID12108	58	남자	도시	24946...	기혼	0	유	무	무	무	무



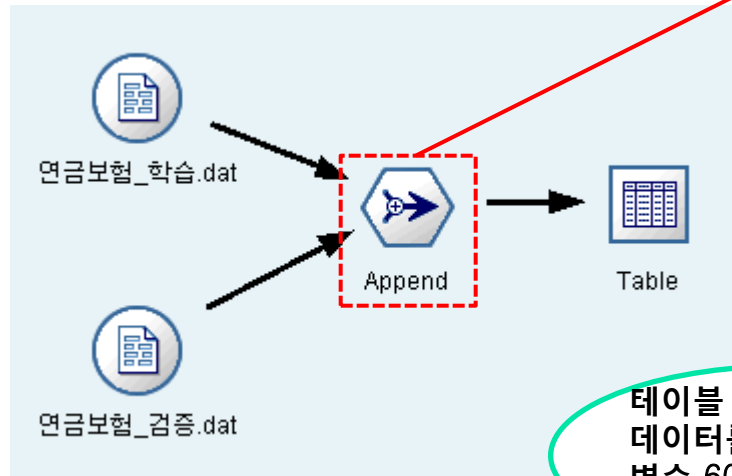
- 연금보험_검증 (12개 변수, 300개 데이터)

This screenshot shows a data table window titled 'Table (12 fields, 300 records) #2'. The table contains 12 columns: 고객번호, 연령, 성별, 거주지역, 소득, 결혼여부, 자녀수, 차량소유, 저축성, 요구불, 모기지론, and 구매여부. The first 8 rows of data are visible, showing various customer profiles.

	고객번호	연령	성별	거주지역	소득	결혼여부	자녀수	차량소유	저축성	요구불	모기지론	구매여부
1	ID12401	19	여자	도심	8162.4...	기혼	1	유	무	무	무	무
2	ID12402	37	여자	도시	15349...	기혼	0	무	무	무	무	무
3	ID12403	45	여자	도시	29231...	기혼	0	무	무	무	무	무
4	ID12404	49	남자	시골	41462...	기혼	3	무	무	무	무	무
5	ID12405	67	여자	시골	57398...	미혼	3	무	무	무	무	무
6	ID12406	35	여자	시골	11520...	기혼	0	무	무	무	무	무
7	ID12407	63	남자	도심	52117...	미혼	2	유	무	무	무	무
8	ID12408	38	남자	시골	26281...	미혼	0	유	무	무	무	무

변수를 기준으로 서로
같은 변수끼리의 결합

Append Node (2/2)



Append

Append 2 datasets

Match fields by: ☐ Position ☒ Name ☐ Match case

Preview of field matches and structure

Output Field	1연금보험_학습.dat	2연금보험_검증.dat
고려번호	고려번호	고려번호
연령	연령	연령
성별	성별	성별
거주지역	거주지역	거주지역
소득	소득	소득
결혼여부	결혼여부	결혼여부
자녀수	자녀수	자녀수
차량소유	차량소유	차량소유
저축성	저축성	저축성
요구불	요구불	요구불
모기지론	모기지론	모기지론
구매여부	구매여부	구매여부

Include fields from: ☒ Main dataset only ☐ All datasets

☐ Tag records by including source dataset in field input

Inputs: Append Annotations

OK Cancel

연금보험_학습에
연금보험_검증 데
이터를 필드를 기
준으로 추가함.

테이블 노드로 결합된
데이터를 확인 (12개
변수, 600(=300+300)
개 데이터)

Table (12 fields, 600 records) #1

	고려번호	연령	성별	거주지역	소득	결혼여부	자녀수	차량소유	저축성	요구불	모기지론	구매여부
1	ID12101	48 여자	도시	17546...	미혼	1 무						
2	ID12102	40 남자	도시	30085...	기혼	3 유						
3	ID12103	51 여자	도시	16575...	기혼	0 유						
4	ID12104	23 여자	도시	20375...	기혼	3 무						
5	ID12105	57 여자	시골	50576...	기혼	0 무						
6	ID12106	57 여자	도시	37869...	기혼	2 무						
7	ID12107	22 남자	시골	8877.0...	미혼	0 무						
8	ID12108	58 남자	도시	24946...	기혼	0 유						
9	ID12109	37 여자	근교	25304...	기혼	2 유						
10	ID12110	54 남자	도시	24212...	기혼	2 유						
11	ID12111	66 여자	도시	59803...	기혼	0 무						
12	ID12112	52 여자	도시	26658...	미혼	0 유						
13	ID12113	44 여자	도시	15735...	기혼	1 무						
14	ID12114	66 여자	도시	55204...	기혼	1 유						
15	ID12115	36 남자	시골	19474...	기혼	0 유						
16	ID12116	38 여자	도시	22342...	기혼	0 유						
17	ID12117	37 여자	도시	17729...	기혼	2 무						
18	ID12118	46 여자	근교	41016...	기혼	0 무						
19	ID12119	62 여자	도시	26909...	기혼	0 무						
20	ID12120	31 남자	도시	22522...	기혼	0 유						
21	ID12121	61 남자	도시	57880...	기혼	2 무						

Table Annotations

OK

Merge Node (1/2)

- 고객_통화 (25개 변수, 3155개 데이터)

실습 DATA

- 고객_통화.dat
- 요금제.dat

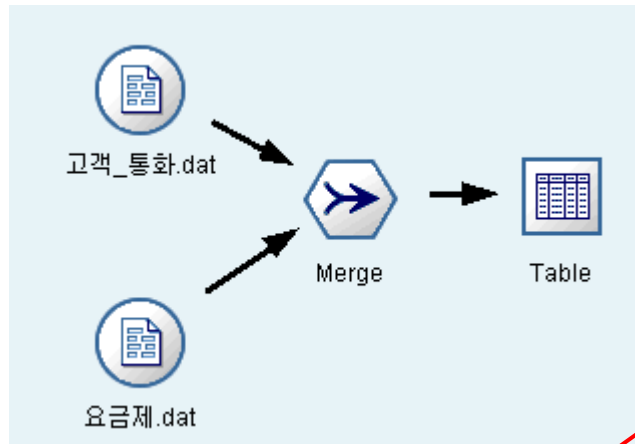


Table (25 fields, 3,155 records)

	고객ID	성별	연령	개시일	서비스기간	단선회수	지불방법	요금제	이탈여부	핸드셋	주간통화회수
1	K100090	남	38	31/05/96	53.167	1	선불	CAT 50	이탈	SOP20	54
2	K100330	남	26	19/02/97	44.367	1	선불	CAT 50	이탈	SOP10	44
3	K100370	남	45	14/06/97	40.533	0	선불	CAT 50	이탈	SOP20	37
4	K100380	남	39	23/12/98	21.967	2	선불	CAT 50	이탈	SOP20	14
5	K100450	남	32	08/08/98	26.533	0	선불	CAT 50	이탈	SOP20	13
6	K100460	남	39	01/08/97	38.933	1	선불	CAT 50	이탈	SOP10	6
7	K100620	남	30	04/03/96	56.100	1	선불	CAT 50	이탈	SOP20	14
8	K100720	남	40	01/02/99	20.633	1	선불	CAT 50	이탈	SOP20	13
9	K100760	남	37	24/05/99	16.900	1	선불	CAT 50	이탈	SOP20	1
10	K100840	남	33	26/05/98	29.000	1	선불	Play 100	이탈	SOP20	25
11	K100940	남	42	05/01/96	56.067	5	선불	Play 100	이탈	SOP10	0
12	K100960	남	25	23/12/97	34.133	0	선불	Play 100	이탈	SOP20	31
13	K101050	남	38	02/01/99	33.800	2	선불	Play 100	이탈	SOP10	0
14	K101060	남	50	02/03/97	41.967	2	선불	Play 100	이탈	SOP10	9
15	K101090	남	37	28/04/97	42.100	5	선불	Play 100	이탈	SOP20	46



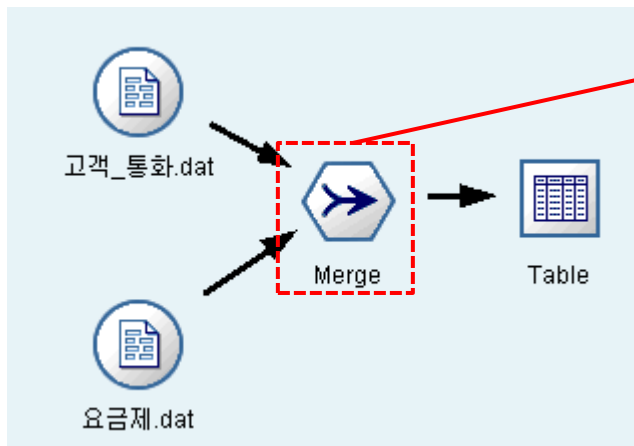
- 요금제 (9개 변수, 5개 데이터)

'요금제'라는 변수가 병합키가 되어 서로 다른 변수끼리의 결합

Table (9 fields, 5 records)

	요금제	기본...	무료통화_분	주간요금_분	야간요금_분	주말요금_분	국제통화요금_분	보이스메일요금	SMS요금
1	Play 100	9.990	100	25	5	5	40	25	15
2	Play 300	12.9...	300	25	2	2	40	15	15
3	CAT 50	13.9...	50	15	5	5	30	10	15
4	CAT 100	17.5...	100	15	5	5	30	10	15
5	CAT 200	25.0...	200	10	5	5	30	10	15

Merge Node (2/2)



Merge

Merge 2 datasets. Merge method: Keys

Merge Method: ☐ Order ☒ Keys

Possible keys:

Keys for merge:

요금제

☒ Combine duplicate key fields

☒ Include only matching records (inner join)

☐ Include matching and non-matching records (full outer join)

☐ Include matching and selected non-matching records (partial outer join)

Select...

☐ Include records in first dataset not matching any others (anti-join)

Inputs Merge Filter Optimization Annotations

OK Cancel Apply Reset

고객_통화 데이터에
요금제 데이터를 '요
금제' 라는 변수를
병합키로 하여 결함.

테이블 노드로 요금제
를 기준으로 병합된 데
이터 확인. (33(=25+9)
개 변수, 3155개 데이
터)

Table (33 fields, 3,155 records) #1

	요금제	고객ID	성별	연령	개시일	서비스기간	단선회수	지불방법	이탈여부	핸드셋	주간통화회수	주간통화시...	야간통화회...
1	CAT 100	K338240	남	28	11/06/96	52.800	1 선불	유지	S50		223	362.400	
2	CAT 100	K296330	여	39	21/10/99	11.900	0 선불	이탈	CAS30		84	751.200	7
3	CAT 100	K165790	여	20	11/01/98	33.500	2 선불	이탈	ASAD90		395	632.400	5
4	CAT 100	K166040	여	14	04/02/97	44.867	5 선불	이탈	ASAD90		274	583.800	16
5	CAT 100	K166150	남	22	14/05/97	41.567	1 선불	이탈	ASAD90		419	741.000	9
6	CAT 100	K166170	여	16	18/05/96	53.600	1 선불	이탈	ASAD90		245	540.000	18
7	CAT 100	K166180	여	22	17/05/99	17.133	0 선불	이탈	ASAD90		319	699.600	6
8	CAT 100	K166400	여	15	03/10/96	49.000	0 선불	이탈	ASAD90		254	459.000	16
9	CAT 100	K167160	여	24	25/11/97	35.067	1 선불	이탈	ASAD90		265	678.600	8
10	CAT 100	K167220	여	17	24/12/98	21.933	0 선불	이탈	S50		197	374.400	1
11	CAT 100	K167310	여	15	01/02/99	20.633	11 선불	유지	ASAD1...		270	708.600	2
12	CAT 100	K167320	여	12	26/06/97	40.133	1 선불	유지	ASAD1...		227	433.800	15
13	CAT 100	K167340	여	24	06/02/96	57.000	1 선불	유지	ASAD1...		242	639.000	3
14	CAT 100	K232910	여	38	27/09/99	12.700	0 선불	이탈	S50		336	517.200	5
15	CAT 100	K232950	여	38	09/09/99	12.700	0 선불	이탈	S50		336	517.200	5

Table Annotations

OK



Sampling

- 무엇을 하는가?
 - 전체 data에서 사용자 필요에 따른 일부 data의 추출
- 왜 하는가?
 - Modeling : Hold-out sample
(test & train & validation)
 - Data 감소로 Performance 향상
 - 불필요한 Data 제거

사용되는 Node



■ 표본(Sample) 노드

- 전체자료에서 일부자료만을 추출을 할 수 있는 노드
- First, 1-in-n, Random% 3가지 방법이 있다.

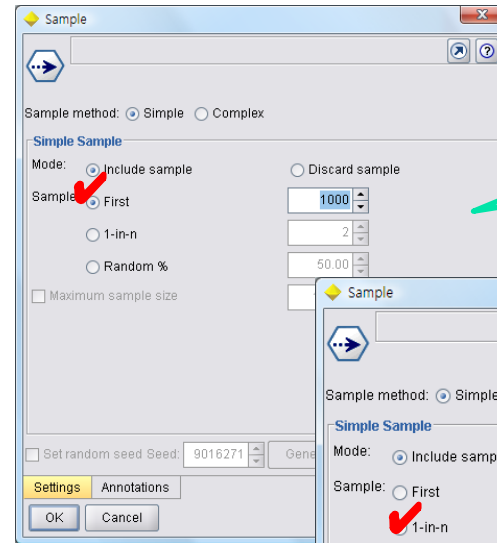
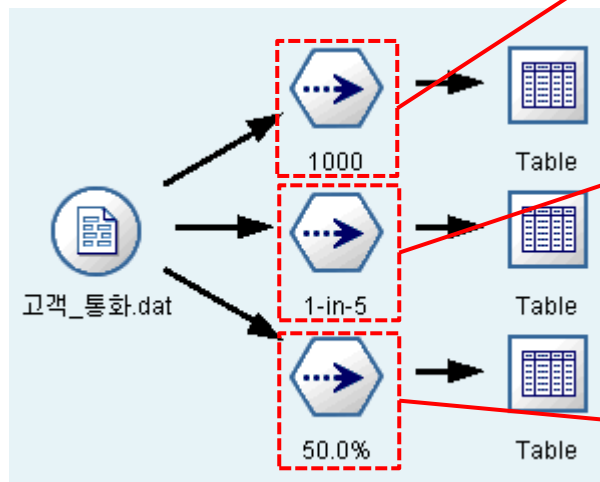


■ 선택(Select) 노드 + 파생(Derive) 노드

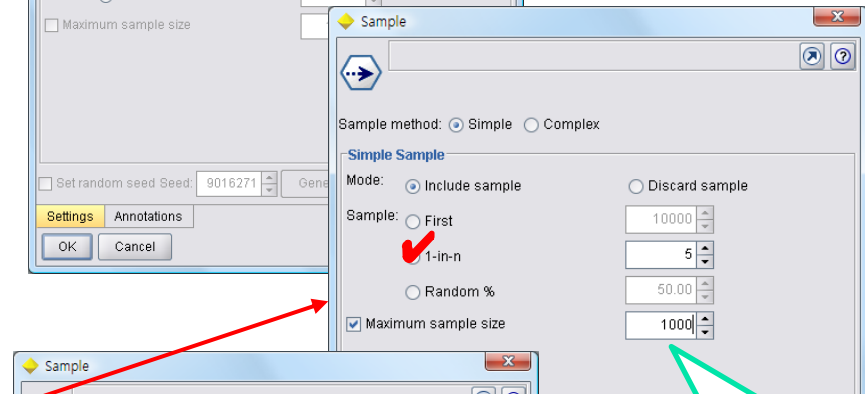
- 사용자정의에 의해서 일부 자료만을 추출할 수 있는 노드
- 연산기호와 CLEM에 의한 방법이 있다.

Sample Node

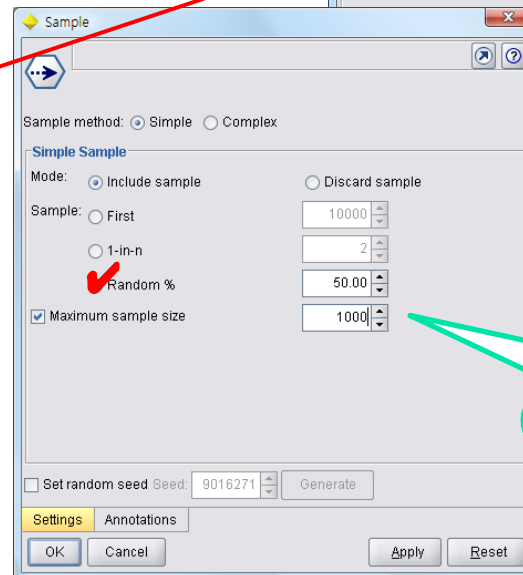
실습 DATA
• 고객_통화.dat



자료의 처음부터
1000개의 레코드를
차례대로 추출



매 5개마다 1개
씩 추출 (최대표
본크기가 1000
을 넘지 않는 레
코드를 추출)



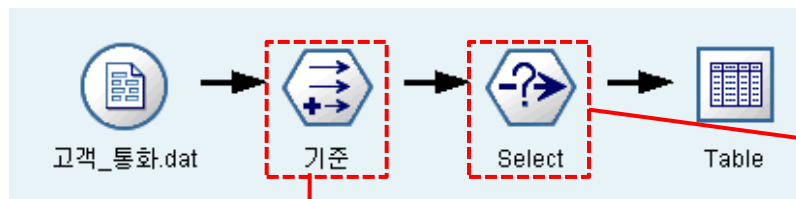
전체의 50%에 해당
되는 레코드를 무작
위로 추출

Select + Derive Node

파생노드와 선택노드를 이용하여 한 data set에서 여러 개의 임의적인 data set을 생성할 수 있다.

실습 DATA

- 고객_통화.dat



Derive as: Formula

Derive field: 기준

Derive as: Formula

Field type: <Default>

Formula: random(3)

Table (26 fields, 3,155...)

	내통화시간	총통화시간	기준
1	1.728	176.649	1
2	2.139	145.538	3
3	2.952	149.568	3
4	3.332	140.829	3
5	2.258	84.090	3
6	3.093	74.343	3
7	3.332	140.829	2
8	9.750	185.355	3
9	8.540	168.101	2
10	1.431	72.434	2
11	0.000	0.000	1
12	1.325	106.489	1
13	0.000	0.000	3
14	1.036	103.044	3
15	2.169	221.343	2
16	2.892	80.612	3
17	1.585	11.511	3
18	2.313	150.038	1
19	3.663	208.558	1
20	1.746	114.320	1

Select

Mode: Include Discard

Condition: 기준=1

Table (26 fields, 3,155...)

	시간	총통화시간	기준
1	.139	145.538	1
2	.332	140.829	1
3	.258	84.090	1
4	.332	140.829	1
5	.540	168.101	1
6	.431	72.434	1
7	.000	0.000	1
8	.325	86.489	1
9	.585	11.511	1
10	.663	208.558	1
11	.746	114.320	1
12	.685	183.648	1
13	.388	203.601	1
14	.529	155.237	1
15	.838	216.115	1
16	.051	69.099	1
17	.177	129.070	1
18	.950	951.894	1
19	.950	1010.393	1
20			

선택 노드를 이용하여 '기준'이라는 필드에서 값이 1인 레코드만 선택함.

파생 노드에서 CLEM 함수 random()을 사용하여 임의적 구분자 1,2,3을 생성함.
기준이라는 필드에 임의적으로 1,2,3이 생성됨.



중복제거와 Balance

- 무엇을 하는가?
 - 자료에서의 중복된 data의 제거와 자료의 균형 맞추기
- 왜 하는가?
 - 효율적인 Modeling 작업을 위해서
 - Data cleansing
(결측값 제거와 함께 대표적인 방법)

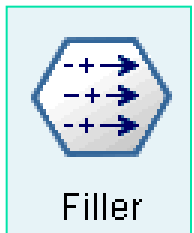
사용되는 Node



- 구별(Distinct) 노드
 - 중복된 자료를 제거하는 기능을 가진 노드



- 균형(Balance) 노드
 - 자료의 균형을 맞추기 위한 노드

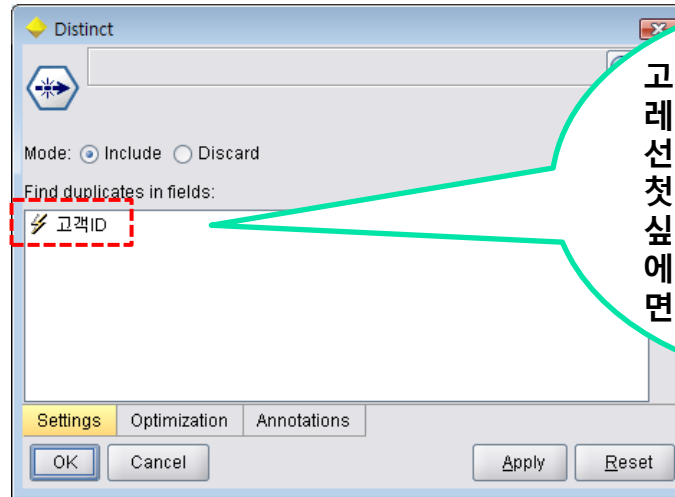
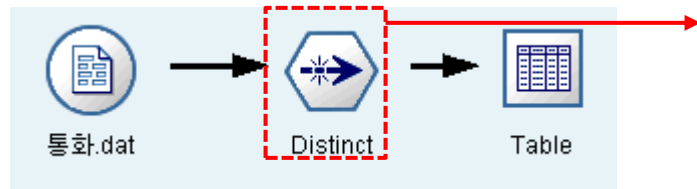


- 채움(Filler) 노드
 - 결측값을 대체하기 위한 노드

Distinct Node

실습 DATA

• 통화.dat



제거 전

Table (10 fields, 19,115 records) #7

	고객ID	주간통화횟수	주간통화...	야간통화횟수	야간통화시간_분	주말통화횟수	주말통화시간_분	국
1	K100010	2	5.494	1	1.212	3	2.210	
2	K100010	2	4.605	1	1.016	3	1.852	
3	K100030	6	10.309	0	0.000	3	5.282	
4	K100030	12	18.597	1	27.600	6	9.529	
5	K100040	8	13.546	0	0.000	4	6.941	
6	K100060	7	6.234	1	1.494	1	1.379	
7	K100070	12	9.148	1	0.651	1	4.012	

제거 후

Table (10 fields, 14,970 records) #3

	고객ID	주간통화횟수	주간통화...	야간통화횟수	야간통화시간_분	주말통화횟수	주말통화시간_분	국제!
1	K100010	2	4.605	1	1.016	3	1.852	
2	K100030	6	10.309	0	0.000	3	5.282	
3	K100040	8	13.546	0	0.000	4	6.941	
4	K100060	7	6.234	1	1.494	1	1.379	
5	K100070	12	9.148	1	0.651	1	4.012	
6	K100100	6	10.671	3	4.797	0	0.000	
7	K100110	7	12.500	3	5.619	0	0.000	

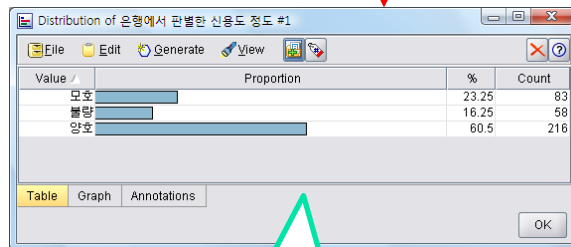
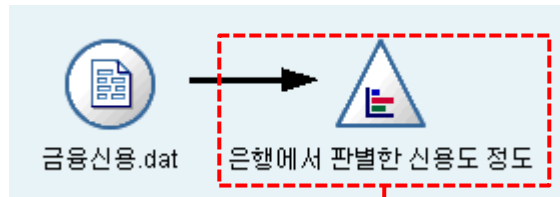
Balance Node

❖ 균형 조정의 목적

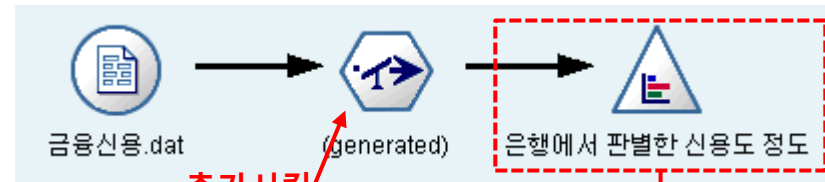
- 효율적인 Modeling 작업 수행
- 목표필드의 분포가 심하게 불균형인 경우
- 인위적으로 균형을 맞출 경우

실습 DATA

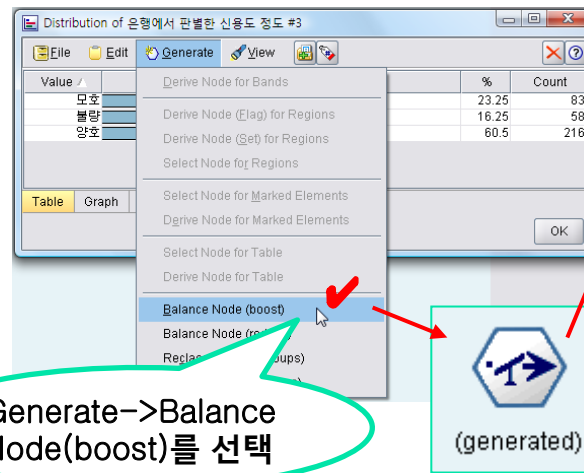
- 금융신용.dat



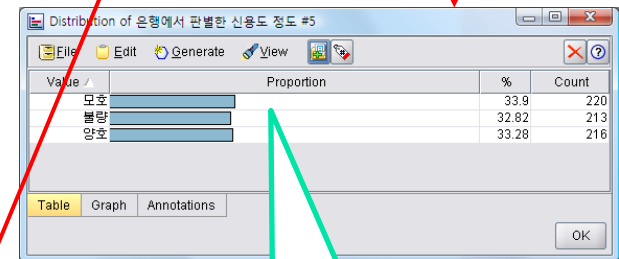
분포노드를 이용하여 신용도의 분포를 살펴봄. 분포에 차이가 있음.



추가시킴



생성됨



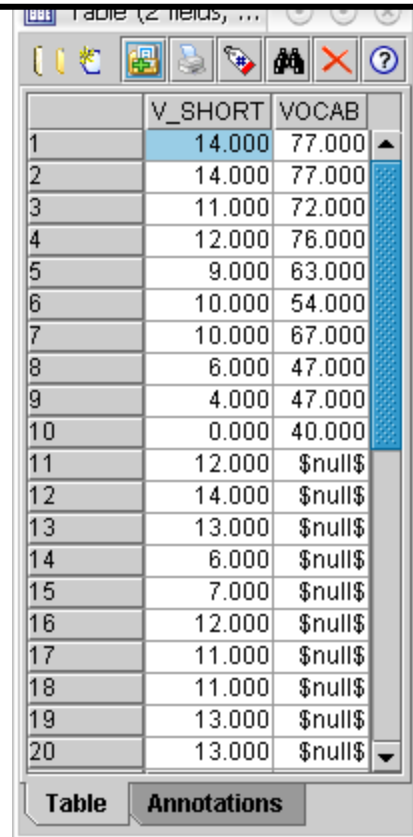
(boost)가장 범주수가 많은 범주 기준으로 나머지 범주의 수가 증가함. (reduce)는 그 반대임.

Filler Node (1/2)

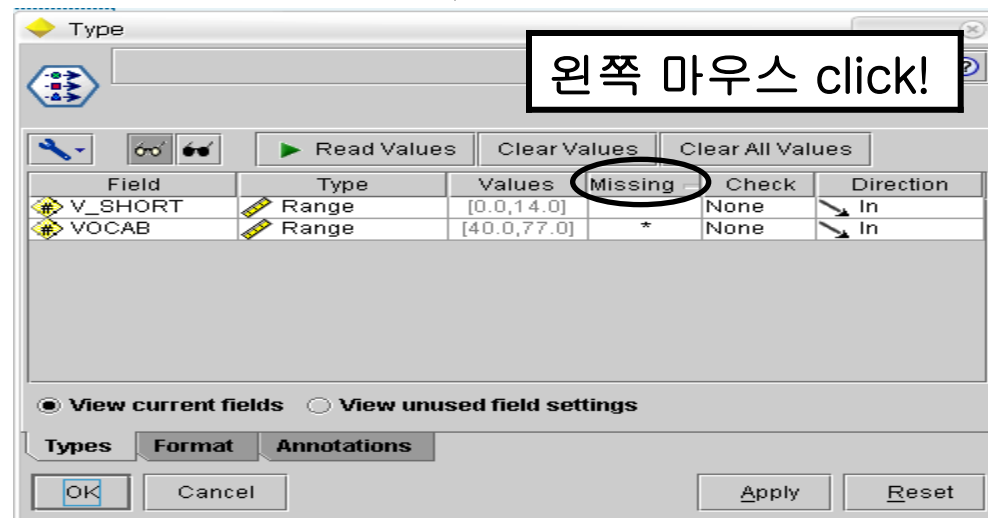
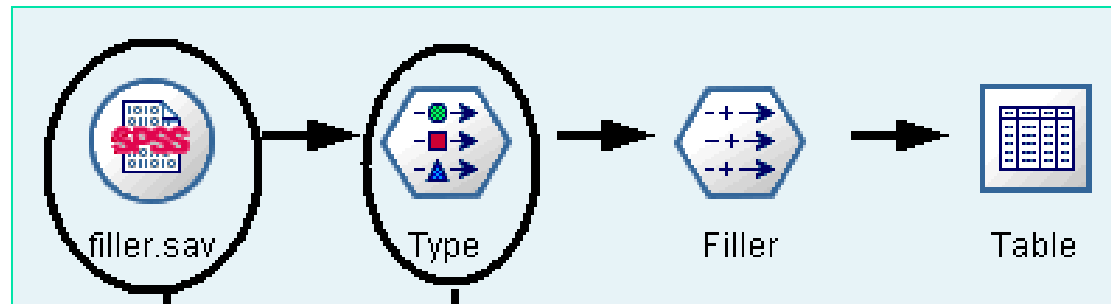
■ 실습 DATA FILE

■ filler.sav

결측값이 있는 data



	V_SHORT	VOCAB
1	14.000	77.000
2	14.000	77.000
3	11.000	72.000
4	12.000	76.000
5	9.000	63.000
6	10.000	54.000
7	10.000	67.000
8	6.000	47.000
9	4.000	47.000
10	0.000	40.000
11	12.000	\$null\$
12	14.000	\$null\$
13	13.000	\$null\$
14	6.000	\$null\$
15	7.000	\$null\$
16	12.000	\$null\$
17	11.000	\$null\$
18	11.000	\$null\$
19	13.000	\$null\$
20	13.000	\$null\$



왼쪽 마우스 click!

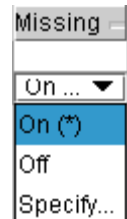
Field	Type	Values	Missing	Check	Direction
# V_SHORT	Range	[0.0,14.0]		None	In
# VOCAB	Range	[40.0,77.0]	*	None	In

☒ View current fields ☐ View unused field settings

Types Format Annotations

OK Cancel Apply Reset

On(*)선택



Missing

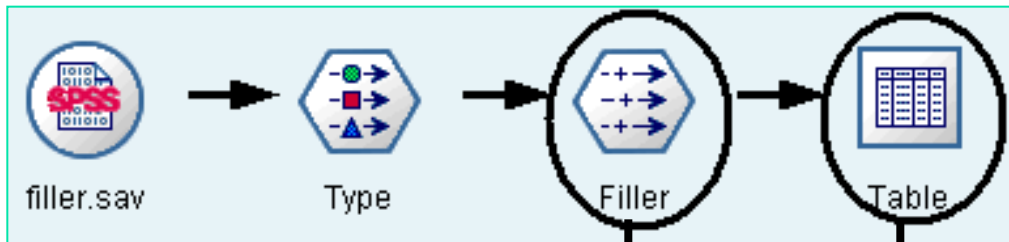
On ... ▼

On (*)

Off

Specify...

Filler Node (2/2)



대체된 결측값

Vocab의
평균값으
로 결측값
을 대체하
는 대화상
자

Filler

Fill in fields:
VOCAB

Replace: Based on condition

Condition:
@BLANK(@FIELD)

Replace with:
@AVE(VOCAB)

Settings Annotations

OK Cancel Apply Reset

Table (2 fields, 4...)

	V_SHORT	VOCAB
1	14.000	77.000
2	14.000	77.000
3	11.000	72.000
4	12.000	76.000
5	9.000	63.000
6	10.000	54.000
7	10.000	67.000
8	6.000	47.000
9	4.000	47.000
10	0.000	40.000
11	12.000	62.000
12	14.000	62.000
13	13.000	62.000
14	6.000	62.000
15	7.000	62.000
16	12.000	62.000
17	11.000	62.000
18	11.000	62.000
19	13.000	62.000
20	13.000	62.000

Table Annotations



Data의 재조정

- 무엇을 하는가?
 - Clementine에서의 기타 data 조정에 관하여 알아본다.
- 왜 하는가?
 - 데이터의 축약과 요약
 - 데이터의 정리
 - 데이터의 또 다른 변환

사용되는 Node



Aggregate

통합(Aggregate) 노드

- 자료를 기준에 맞추어 통합하여 산출하는 노드 (SQL의 Group by 기능과 동일)



Sort

정렬(Sort) 노드

- 자료를 기준에 맞추어 정렬하는 노드 (SQL의 Order by 기능과 동일)



SetToFlag

이분변환(Set to flag) 노드

- 필드의 조합을 이진 자료(T or F)로 변환시키는 노드

Aggregate Node

실습 DATA

- 통화.dat



Table (10 fields, 19,115 records)

	고객ID	주간통화횟수	주간통화...	야간통화...	야간통화...	주말통화...	주말통화...	국제통화...	국제통화...
1	K100290	11	17.550	1	8.892	4	7.507	2.053	
2	K100300	7	14.004	1	4.844	2	4.796	1.951	
3	K100340	6	11.904	0	0.000	2	4.077		
4	K100380	2	11.339	3	6.185	0	0.000		
5	K100550	3	12.150	2	5.006	0	0.000		
6	K100610	0	0.000	1	2.545	0	0.000		
7	K100780	2	5.126	0	0.000	0	0.000		
8	K100870	5	4.333	4	2.600	0	0.000		
9	K101030	5	8.441	3	9.496	0	0.000		
10	K101260	2	7.335	6	13.064	1	2.885		
11	K101270	0	0.000	2	26.435	0	0.000		
12	K101280	5	8.113	1	8.850	2	4.127		
13	K101360	9	14.463	4	11.364	2	6.543		
14	K101390	4	9.095	6	9.815	0	0.000		
15	K101440	1	2.054	6	14.111	0	0.000		
16	K101450	3	8.731	2	6.035	1	7.704		
17	K101590	1	2.883	6	10.209	1	2.598		
18	K101620	0	0.000	2	8.724	1	3.750		
19	K101830	1	5.614	6	13.753	0	0.000		
20	K101840	0	0.000	0	0.000	1	10.055		

Aggregate

Key fields: 고객ID

Aggregate fields:

Field	Sum	Mean	Min	Max	SDev
주간통화횟수	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
주간통화시간_분	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
야간통화횟수	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
야간통화시간_분	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
주말통화횟수	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
주말통화시간_분	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
국제통화요금_분	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Default mode: ☒ Sum ☒ Mean ☐ Min ☐ Max ☐ SDev

New field name extension: Add as: ☒ Suffix ☐ Prefix

☐ Include record count in field Record_Count

Settings Annotations

OK Cancel Apply Reset

고객ID를 기준으로
통합하고 싶은
변수를 선택함.

테이블 노드로
통합된 결과를
확인

Table (9 fields, 14,970 records)

	고객ID	주간통화횟수_Sum	주간통화시간_분_Sum	야간통화횟수_Sum	야간통화시간_분_Sum	주말통화...	주말통화...
1	K417620	9	68.069	23	53.444		
2	K417160	13	71.389	34	105.629		
3	K417100	16	86.374	33	111.379		
4	K417080	16	62.115	28	98.484		
5	K417020	12	83.470	76	116.470		
6	K416960	12	67.061	18	87.012		
7	K416830	2	19.538	26	53.248		
8	K416800	9	32.885	45	105.963		
9	K416680	25	77.906	41	68.613		
10	K416510	30	58.064	23	95.944		
11	K416380	14	19.929	29	77.483		
12	K416330	52	102.721	37	113.458		
13	K416200	38	57.074	15	67.446		
14	K415920	6	8.626	8	66.216		
15	K415890	28	48.899	47	100.127		
16	K414080	3	8.815	4	7.658		
17	K413860	4	11.475	27	60.514		
18	K413640	7	7.018	4	7.993		
19	K413250	21	54.262	1	367.800		
20	K413030	9	8.269	12	12.217		

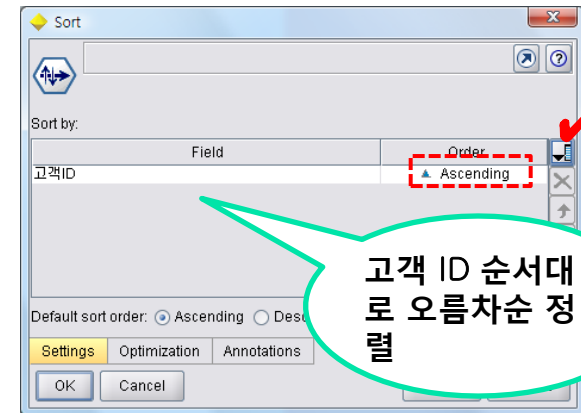
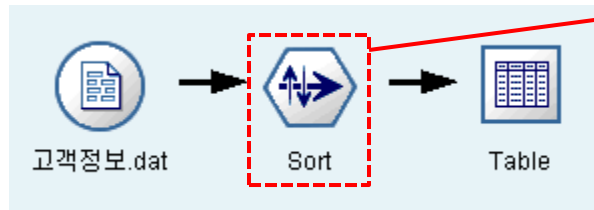
■ 통합(Aggregate) 노드

: 합계, 평균, 최소값, 최대값 등의 통합함수를 적용하여 파생 데이터를 만든다.

Sort Node

실습 DATA

- 고객정보.dat



테이블 노드로
ID를 기준으로
정렬 확인

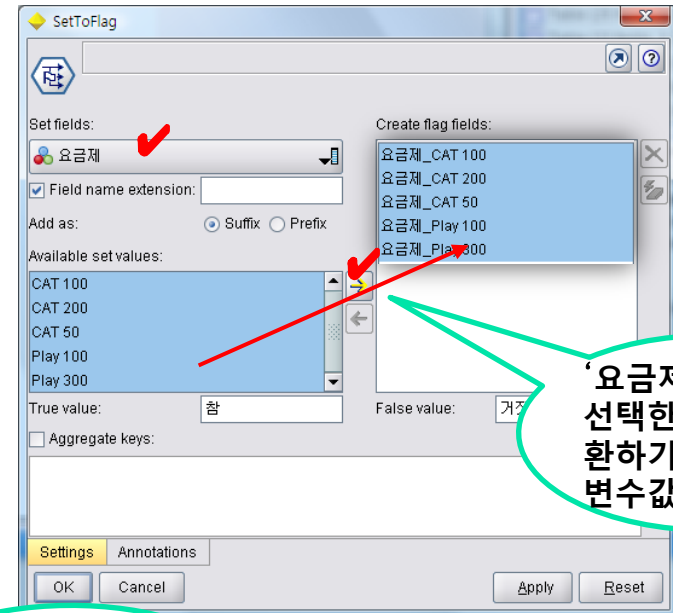
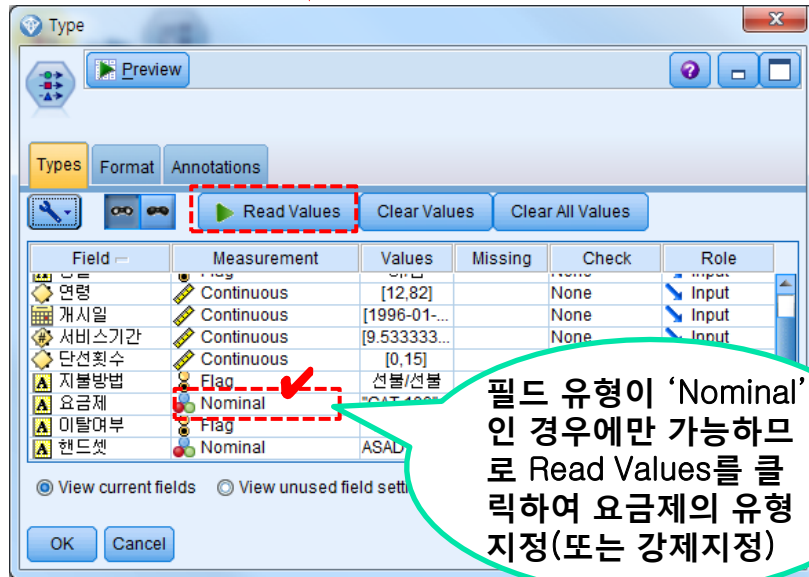
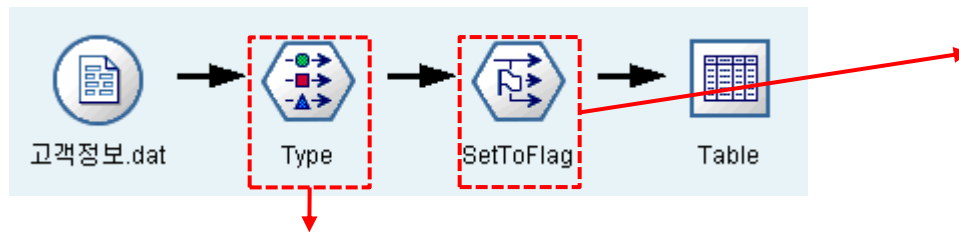
Table (10 fields, 31,769 records) #1

	고객ID	성별	연령	개시일	서비스기간	단선회수	지불방법	요금제
1	K100010	남	46	18/07/99	15.067	1	선불	CAT 50
2	K100020	남	27	18/09/97	37.333	0	선불	CAT 50
3	K100030	남	39	23/08/96	50.367	2	선불	CAT 50
4	K100040	남	28	18/08/98	26.200	2	선불	CAT 50
5	K100050	남	47	11/08/98	26.433	0	선불	CAT 50
6	K100060	남	29	06/07/97	39.800	1	선불	CAT 50
7	K100070	남	38	14/10/99	12.133	1	선불	CAT 50
8	K100080	남	27	20/01/98	33.200	0	선불	CAT 50
9	K100090	남	38	31/05/96	53.167	1	선불	CAT 50
10	K100100	남	48	28/07/96	51.233	0	선불	CAT 50
11	K100110	남	34	11/12/98	22.367	2	선불	CAT 50
12	K100120	남	32	06/11/97	35.700	0	선불	CAT 50
13	K100130	남	45	15/12/96	46.567	2	선불	CAT 50
14	K100140	남	29	24/03/99	18.933	1	선불	CAT 50
15	K100150	남	30	28/07/98	26.900	2	선불	CAT 50
16	K100160	남	28	18/07/99	15.067	9	선불	CAT 50
17	K100170	남	26	28/06/97	40.067	1	선불	CAT 50
18	K100180	남	37	26/06/98	27.967	1	선불	CAT 50
19	K100190	남	38	06/08/97	38.767	2	선불	CAT 50
20	K100200	남	46	09/10/96	48.800	0	선불	CAT 50

Set To Flag Node

실습 DATA

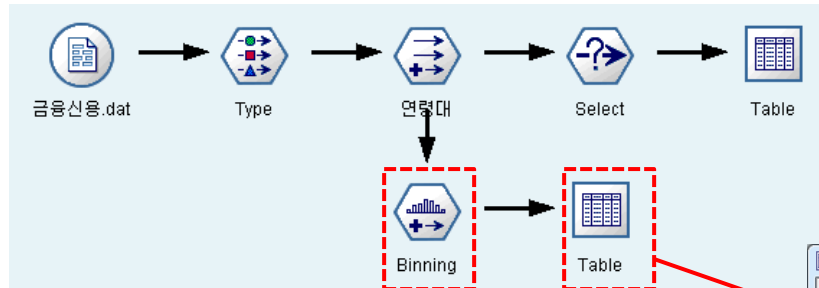
- 고객정보.dat



이분형 변환된 데이터 확인

	요금제	이탈여부	핸드셋	요금제_CAT 100	요금제_CAT 200	요금제_CAT 50	요금제_Play 100	요금제_Play 300
1	CAT 50	이탈	SOP10	거짓	거짓	참	거짓	거짓
2	CAT 50	이탈	SOP10	거짓	거짓	참	거짓	거짓
3	CAT 50	이탈	SOP20	거짓	거짓	참	거짓	거짓
4	CAT 50	이탈	SOP10	거짓	거짓	참	거짓	거짓
5	CAT 50	이탈	SOP10	거짓	거짓	참	거짓	거짓
6	CAT 50	이탈	SOP10	거짓	거짓	참	거짓	거짓
7	CAT 50	이탈	SOP20	거짓	거짓	참	거짓	거짓
8	CAT 50	이탈	SOP20	거짓	거짓	참	거짓	거짓
9	CAT 50	이탈	SOP20	거짓	거짓	참	거짓	거짓
10	CAT 50	이탈	SOP20	거짓	거짓	참	거짓	거짓
11	CAT 50	이탈	SOP20	거짓	거짓	참	거짓	거짓
12	CAT 50	이탈	SOP20	거짓	거짓	참	거짓	거짓
13	CAT 50	이탈	SOP20	거짓	거짓	참	거짓	거짓

데이터 구간화



❖ 구간화의 필요성

- 연속형 데이터의 범주화
- 범주화를 통해서 정확한 의미 부여
- 각종 분위수의 산정을 통한 정보의 이해 증진

❖ 구간화(Binning) 노력

연속형 변수를 등간격, 등비율, 평균/표준편차 등을 고려한 형태의 구간화 규칙을 적용하여 범주형 변수로 변환

Binning Dialog Box Settings:

- Bin fields: 월평균수입(1000원)
- Binning method: Tiles (equal count) ✓
- Equal Count Binning:
 - Tile name extension: _TILE (Add as: Suffix)
 - Custom tile extension: _TILEN (Add as: Suffix)
 - ☒ Quartile (4)
 - ☒ Decile (10)

Resulting Table (Table 15 fields, 357 records):

미 된 년수	현 직장에서 근무한 년수	은행에서 판별한 신용도 정도	연령대	월평균수입(1000원)_TILE4	월평균수입(1000원)_TILE10
1	5	5 양호	20대 이하	2	3
2	3	6 모호	30대	4	9
3	7	2 모호	20대 이하	2	3
4	8	9 양호	40대 이상	4	9
5	10	10 양호	40대 이상	4	10
6	3	2 양호	20대 이하	1	3
7	4	4 양호	40대 이상	3	6
8	3	4 양호	30대	3	6
9	4	5 양호	40대 이상	3	7
10	4	8 양호	30대	3	6
11	6	8 모호	20대 이하	4	9

Binned field details (from Binning dialog):

Bin	Lower	Upper
1	>= 138	< 349
2	>= 349	< 421
3	>= 421	< 506
4	>= 506	<= 923

월평균수입 변수를 4 분위수와 10분위수 두가지로 구간화함

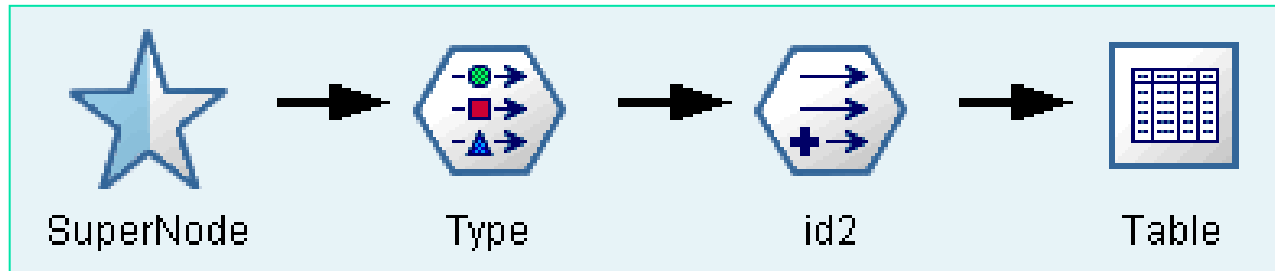


Super Node

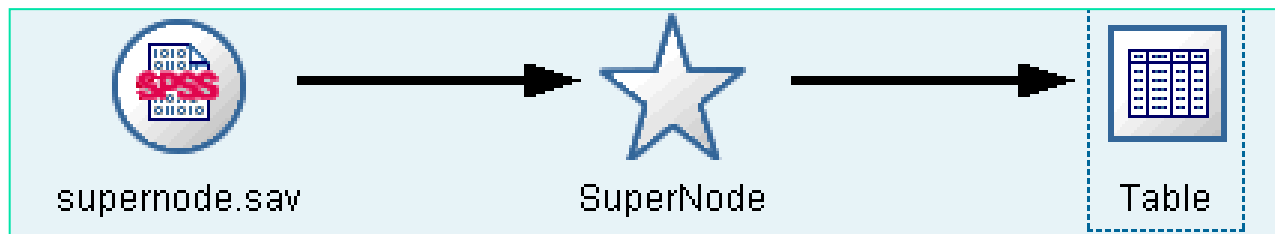
- 무엇을 하는가?
 - 노드의 압축 노드인 슈퍼노드의 생성
- 왜 하는가?
 - 수 많은 노드의 통합
 - 일정한 process 노드를 고정시켜 다른 Stream에서도 활용 (저장후 load)
- 어디에서?
 - 모형화 완료 후 Stream 정리時 주로 사용

Super Node의 종류

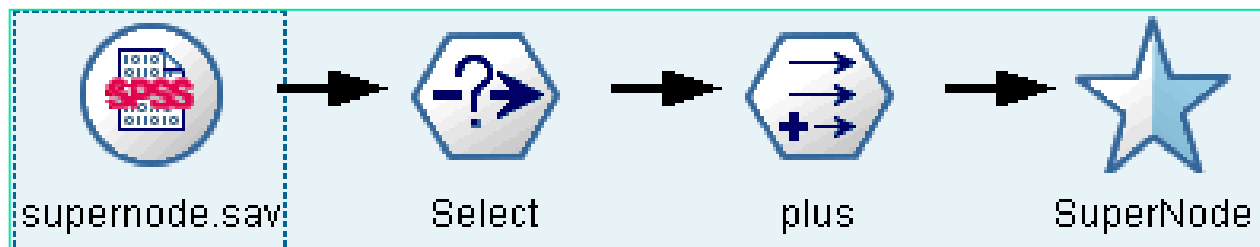
- Source Supernode: 슈퍼노드에 Source 노드를 포함한 경우



- Manipulation Supernode: Source 노드와 Output노드를 포함하지 않은 경우



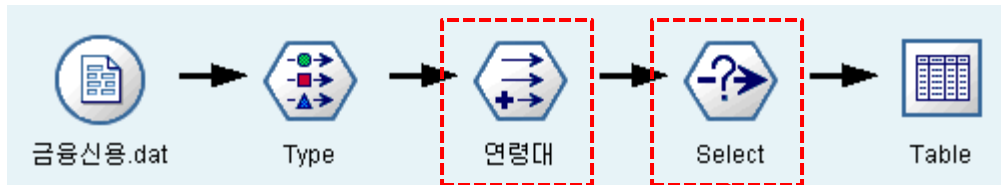
- Terminal Supernode: 슈퍼노드에 Output노드를 포함한 경우



Super Node의 생성 (1/2)

실습 DATA

• 금융신용.dat



연령대

Derive as: Set

Mode: ☒ Single ☐ Multiple

Derive field: 연령대

Derive as: Set

Field type: Set

Default value: default

Set field to	If this condition is true
20대이하	연령 <= 29
30대	연령 >= 30 and 연령 <= 39
40대이상	연령 >= 40

Settings Annotations

OK Cancel Apply Reset

Select

Mode: ☒ Include ☐ Discard

Condition: 연령대="30대"

Settings Annotations

OK Cancel Apply Reset

다음과 같은 스트림을 생성.
파생 노드와 선택 노드를 하나의 수퍼 노드로 압축시킬 것임.

Super Node의 생성 (2/2)

수퍼노드가
생성됨.

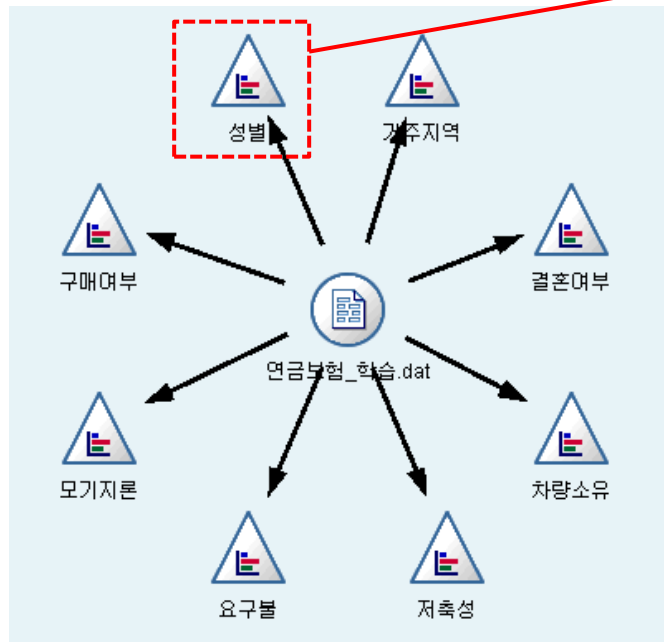
두 노드를 선택한 후
마우스 오른쪽 버튼
을 눌러 Create
SuperNode를 클릭
함

수퍼노드 해제시

수퍼노드를 해제
하고 싶을 경우
Expand를 클릭하
고 수퍼노드를 편
집하고 싶을 경우
Zoom In을 클릭
함.

그래프를 이용한 데이터 탐색 (1/2)

범주형 데이터 탐색



성별

Field: 성별

Plot: ☒ Selected fields ☐ All flags (true values)

Field: ☒ 성별

Overlay

Color: ☒ 구매여부

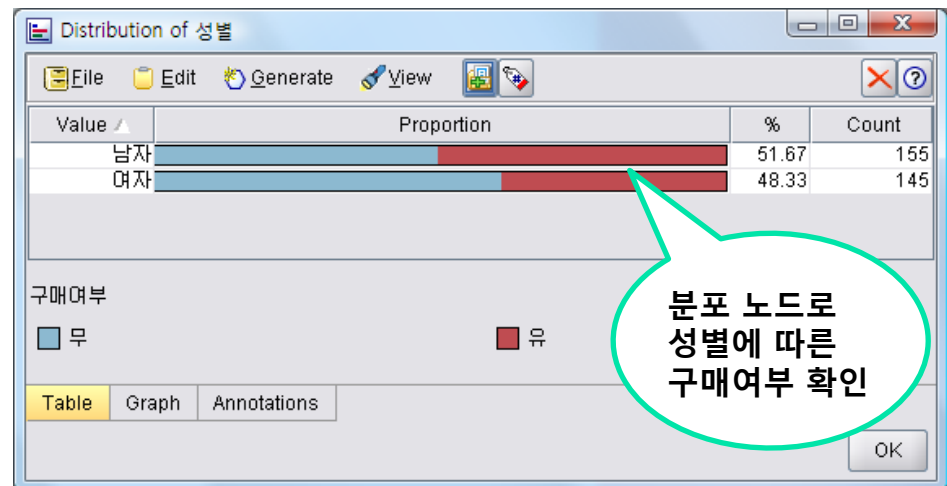
☒ Normalize by color

Sort: ☒ Alphabetic ☐ By count

☐ Proportional scale

Plot Appearance Output Annotations

OK Execute Cancel Apply Reset

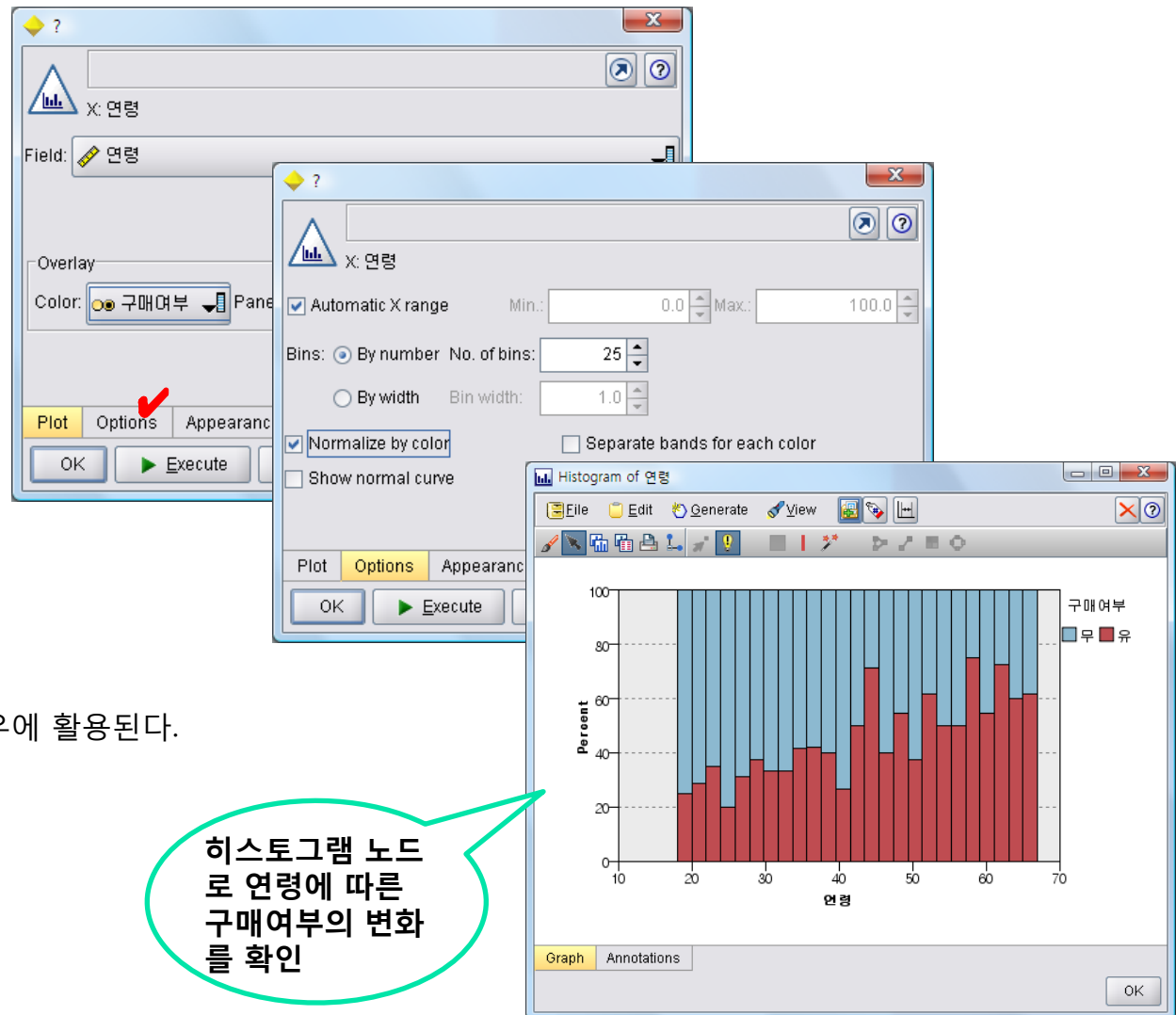
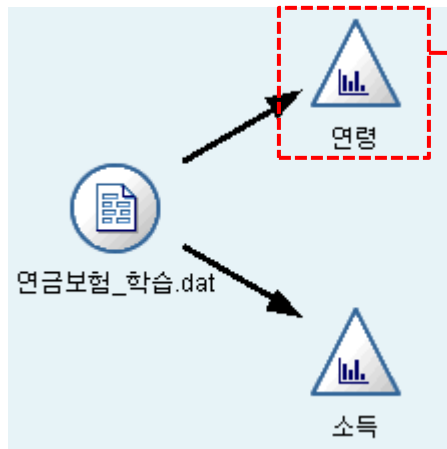


■ 분포(Distribution) 노드

: 범주형 데이터 자료를 탐색할 때 활용된다.

그래프를 이용한 데이터 탐색 (2/2)

연속형 데이터 탐색

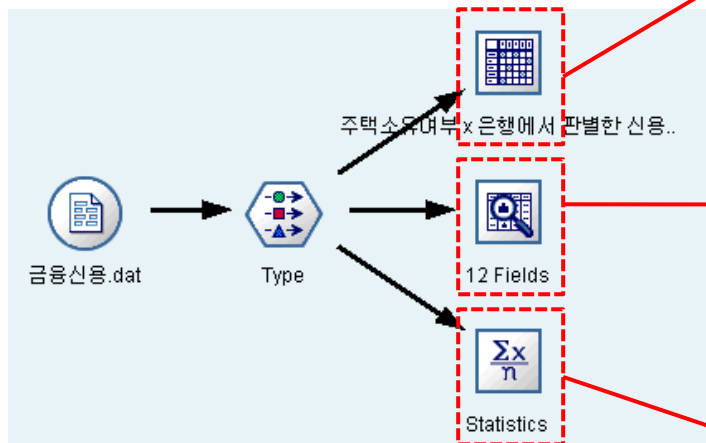


- 히스토그램(Histogram) 노드
: 연속형 데이터의 탐색을 하는 경우에 활용된다.

데이터의 다양한 결과 분석

❖ 다양한 결과 분석의 필요성

- 최종 output의 출력
- 데이터의 최종 결과 값 도출 및 분석
- 결과의 확인 작업



Matrix of 주택소유여부 by 은행에서 판별한 신용..

은행에서 판별한 신용도 정도

주택소유여부	모호	불량	양호
미소유	60	43	159
소유	23	15	57

Cells contain: cross-tabulation of fields (including missing values)
Chi-square = 0.074, df = 2, probability = 0.964

Data Audit of [12 fields] #1

Field	Graph	Type	Min	Max	Mean	Std. Dev	Skewness
고객의 일련번호		Range	111982034	897257027	501429567.622	233981025.489	0.054
A 결혼상태		Flag	--	--	--	--	--
A 신용카드 사용..		Flag	--	--	--	--	--
A 주택소유여부		Flag	--	--	--	--	--
월평균집세(10..		Range	--	--	--	--	--

Statistics of [6 fields]

월평균집세(1000원)

Statistics	Value
Count	357
Mean	280.255
Min	0
Max	2473
Range	2473
Variance	117346.769
Standard Deviation	342.559
Standard Error of Mean	18.130

월평균수입(1000원)

Statistics	Value
Count	357
Mean	436.829
Min	138
Max	923
Range	785
Variance	23080.041
Standard Deviation	151.921
Standard Error of Mean	9.044

▪ 교차표(Matrix) 노트

: 범주형 자료의 교차 분석을 한다. (Chi-2 검정 가능)

▪ 데이터 검토(Data Audit) 노트

: 전체 필드의 그래프와 탐색적 기술 통계량을 보여준다.

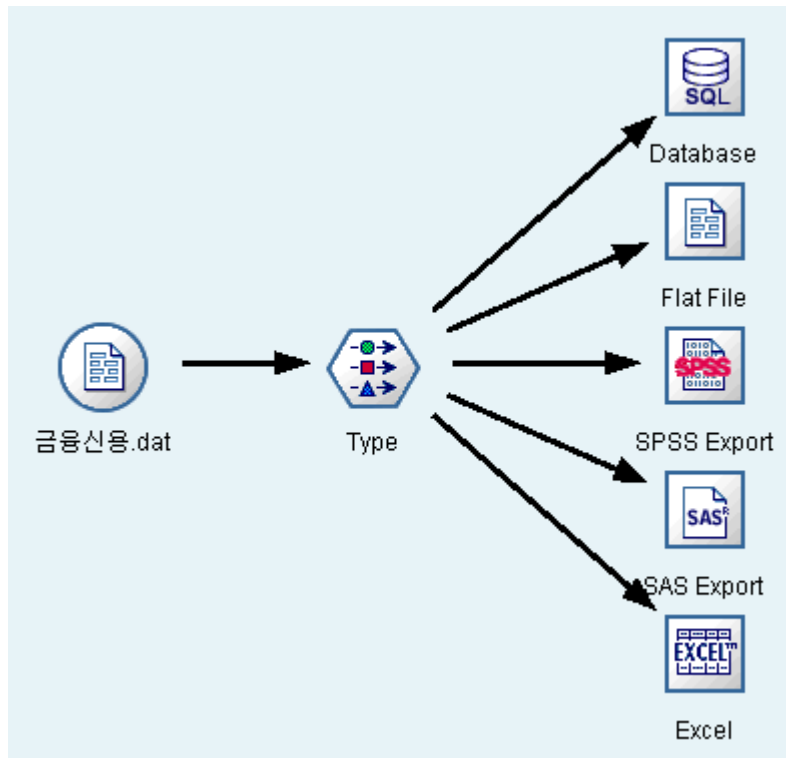
▪ 통계량(Statistics) 노트

: 각종 기술 통계량을 전문적으로 수행한다.

데이터의 다양한 출력

❖ 출력의 필요성

- 결과의 저장 작업
- 결과의 파일화 작업
- 다른 시스템에서 활용 가능한 형태로의 변환



▪ 데이터베이스(Database) 노드

: ODBC(Open Database Connectivity)를 이용하여, 데이터베이스 테이블을 생성 또는 삽입할 수 있다.

▪ 플랫 파일(Flat File) 노드

: 출력 결과를 구분자에 의한 텍스트 파일로 저장하는 기능을 한다.

▪ SPSS 출력(SPSS Export) 노드

: '*.sav' 형식으로 데이터를 내보내는 기능을 한다. SPSS 제품에서 불러올 수 있다.

▪ SAS 출력(SAS Export) 노드

: '*.sd2', '*.SSD*', '*.sas7bdat' 형식으로 데이터를 내보내는 기능을 한다. SAS 제품에서 불러올 수 있다.

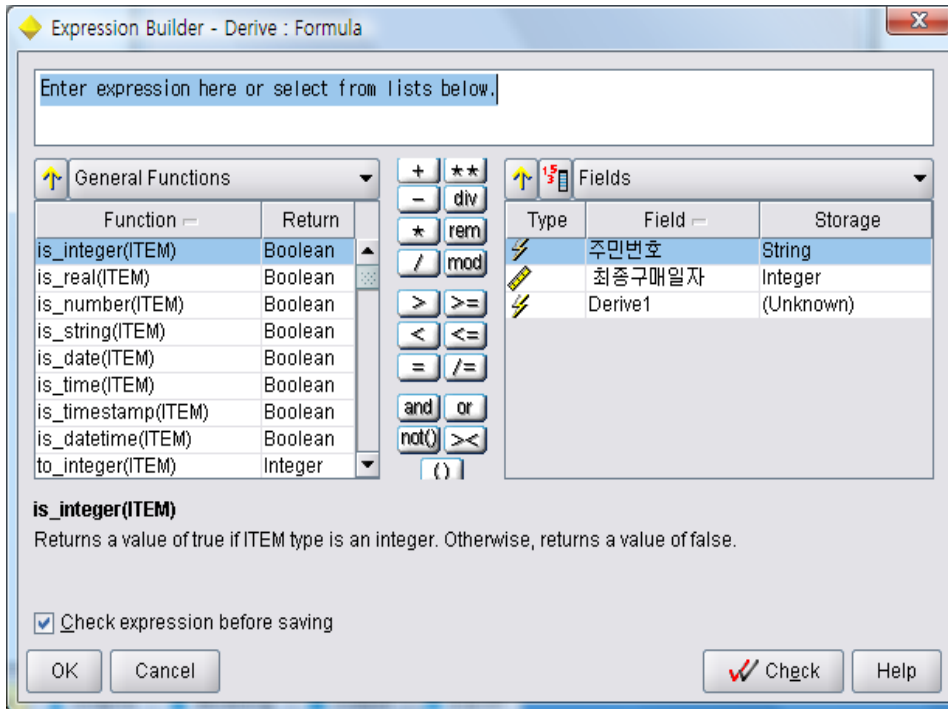
▪ EXCEL 노드

: 데이터를 내보내고 Microsoft Excel에서 파일을 여는 기능을 제공한다.

CLEM 언어 (1/3)

CLEM(Clementine Language for Expression Manipulation)

CLEM 언어는 Clementine 스트림 내에서 데이터의 변환 및 정제 등의 기능을 하는 함수를 의미하며 CLEM은 Derive, Select, Filter, Balance, Report 노드에 사용할 수 있다.

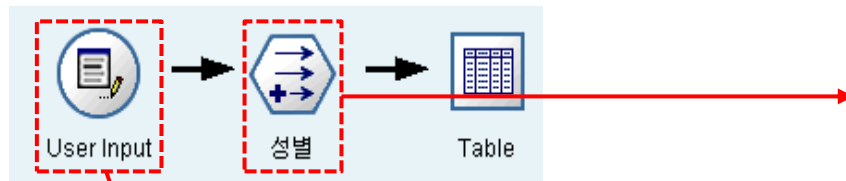


❖ 자주 사용하는 CLEM 함수

- **substring(N, LEN, STRING)**
: 특정 필드의 N번째 위치로부터 LEN만큼의 문자열을 반환
- **date_days_difference(DATE1, DATE2)**
: 날짜1에서 날짜2까지 시간(일)을 정수로 반환
- **@OFFSET(FIELD, EXPR)**
: 특정 필드로부터 EXPR만큼의 이전 레코드 필드값을 반환
- **Field1 > < Field2**
: 두 개의 필드값을 연결시켜 하나의 필드값으로 반환
- **to_string(ITEM)**
: 필드의 유형을 문자열로 변경
- **@INDEX**
: index의 생성

CLEM 언어 (2/3)

CLEM 언어를 이용한 새로운 변수 생성



User Input (2 fields, 4 records)

Field	Storage	Values
주민번호	String	"800707-1234567" "860510-2556487"
최종구매일자	String	"20080501" "20071224"

Generate data: ☒ All combinations ☐ In order

Data Filter Types Annotations

OK Cancel

User Input 노드를 사용하여 데이터를 직접 입력

Derive

Derive as: Conditional

Mode: ☒ Single ☐ Multi

Derive field: 성별

Derive as: Conditional ☒

Field type: Set

If: substring(8,1, 주민번호)="1" ☒

Then: 남

Else: 여

Settings Annotations

OK Cancel

Clem 언어 중에 하나인 substring을 사용하여 주민번호 문자열의 8번째 자리가 '1'이면 '남자' 아니면 '여자'인 성별 변수를 생성

Table (3 fields, 4 records)

	주민번호	최종구매	성별
1	800707-1234567	20080501	남
2	800707-1234567	20071224	남
3	860510-2556487	20080501	여
4	860510-2556487	20071224	여

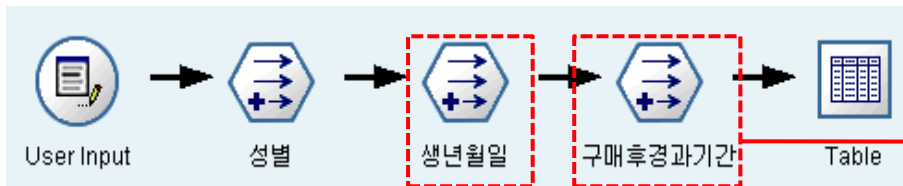
File Edit Generate

OK

테이블 노드로 생성된 성별 변수 확인

CLEM 언어 (3/3)

CLEM 언어를 이용한 새로운 변수 생성



‘substring’을 사용하여 주민번호 문자열의 처음부터 6 자리를 추출한 후 ‘>’를 사용하여 ‘19’와 연결시킴

Derive

Derive as: Formula

Mode: ☒ Single ☐ Multiple

Derive field: 생년월일

Derive as: Formula

Field type: <Default>

Formula: "19" > substring(1,6, 주민번호)

Settings Annotations

OK Cancel Apply Reset

‘date_days_difference’를 사용하여 최종구매일자부터 현재까지 경과한 일수를 반환

Derive

Derive as: Formula

Mode: ☒ Single ☐ Multiple

Derive field: 구매후경과기간

Derive as: Formula

Field type: <Default>

Formula: date_days_difference(최종구매일자, @TODAY)

Settings Annotations

OK Cancel

Table (5 fields, 4 records)

	주민번호	최종구매...	성별	생년월일	구매후경과기간
1	800707-1234567	20080501	남	19800707	291
2	800707-1234567	20071224	남	19800707	420
3	860510-2556487	20080501	여	19860510	291
4	860510-2556487	20071224	여	19860510	420

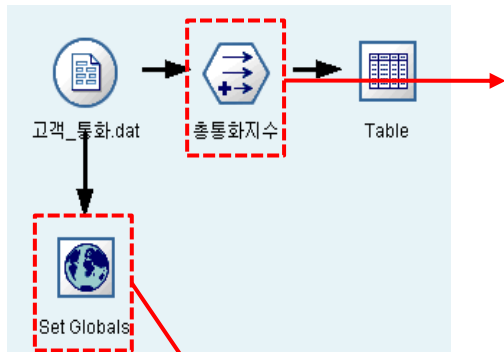
Table Annotations

OK

전역값 설정

실습 DATA

- 고객_통화.dat



Derive as: Formula

Mode: ☒ Single ☐ Multiple

Derive field: 총통화지수

Derive as: Formula

Field type: <Default>

Formula: 총통화시간_분 / @GLOBAL_MEAN(총통화시간_분)

전역값(Filler) 노드

: 필드의 통합 계산된 값을 memory에 저장하여 스트림내에서 자유롭게 사용할 수 있게 한다.

설정한 전역값을 파생 노드를 통해 새로운 변수 생성에 사용함

Globals to be created:

Field	MEAN	SUM	MIN	MAX	SDEV
총통화시간_분	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Default operation(s): ☒ MEAN ☒ SUM ☒ MIN ☒ MAX ☒ SDEV

☒ Clear all globals before executing

☐ Display preview of globals created after execution

Settings Annotations

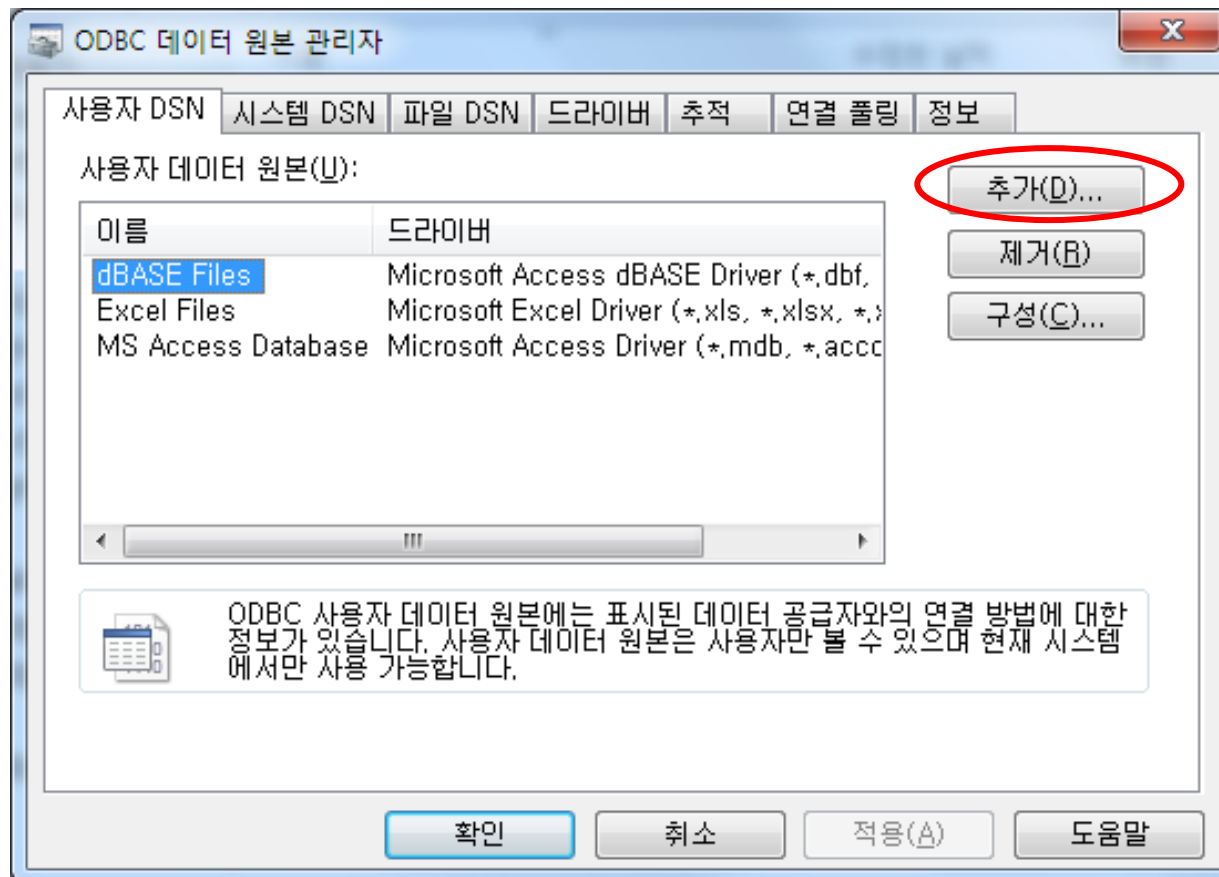
OK Execute Cancel Apply Reset

총통화시간_분의 전역값을 설정함

회수	국내통화시...	평균국내통화시간	총통화시간	총통화지수
1	88	152.100	1.728	0.148
2	56	119.758	2.139	0.122
3	50	147.600	2.952	0.125
4	38	126.600	3.332	0.118
5	35	79.029	2.258	0.070
6	24	74.222	3.093	0.062
7	38	126.600	3.332	0.118
8	18	175.500	9.750	0.155
9	15	128.100	8.540	0.140
10	45	64.400	1.431	0.061
11	0	0.000	0.000	0.000
12	66	87.450	1.325	0.089
13	0	0.000	0.000	0.000
14	85	88.092	1.036	0.086
15	78	169.200	2.169	0.185
16	25	72.311	2.892	0.067
17	1	1.585	1.585	0.010
18	54	124.900	2.313	0.125
19	43	157.500	3.663	0.174
20	65	113.488	1.746	0.096

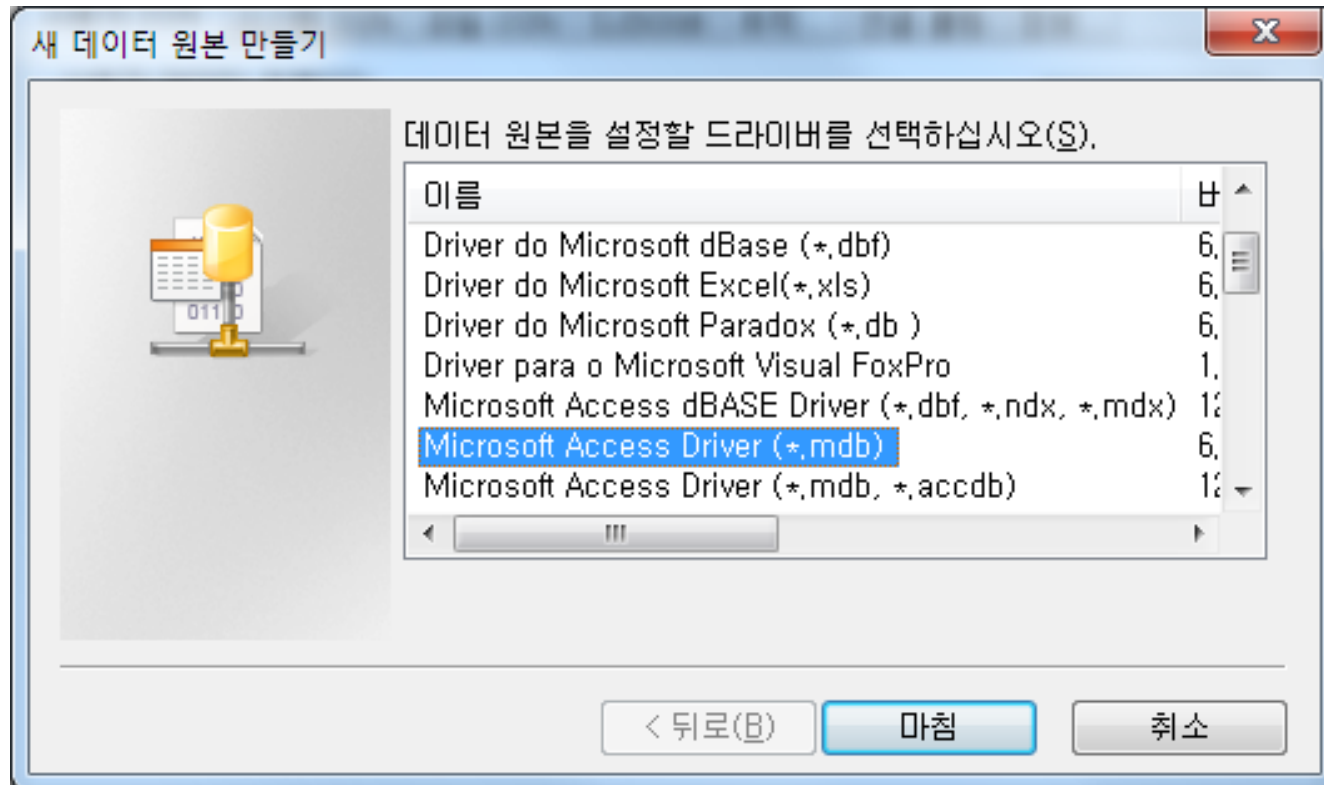
ODBC 설정 (1/4)

- 윈도우의 "시작">"프로그램 및 파일 검색"에서
C:\Windows\SysWOW64\odbcad32.exe 입력 후 실행
- 아래와 같이 "사용자 DSN" 탭을 선택한 후 "추가" 버튼을 클릭



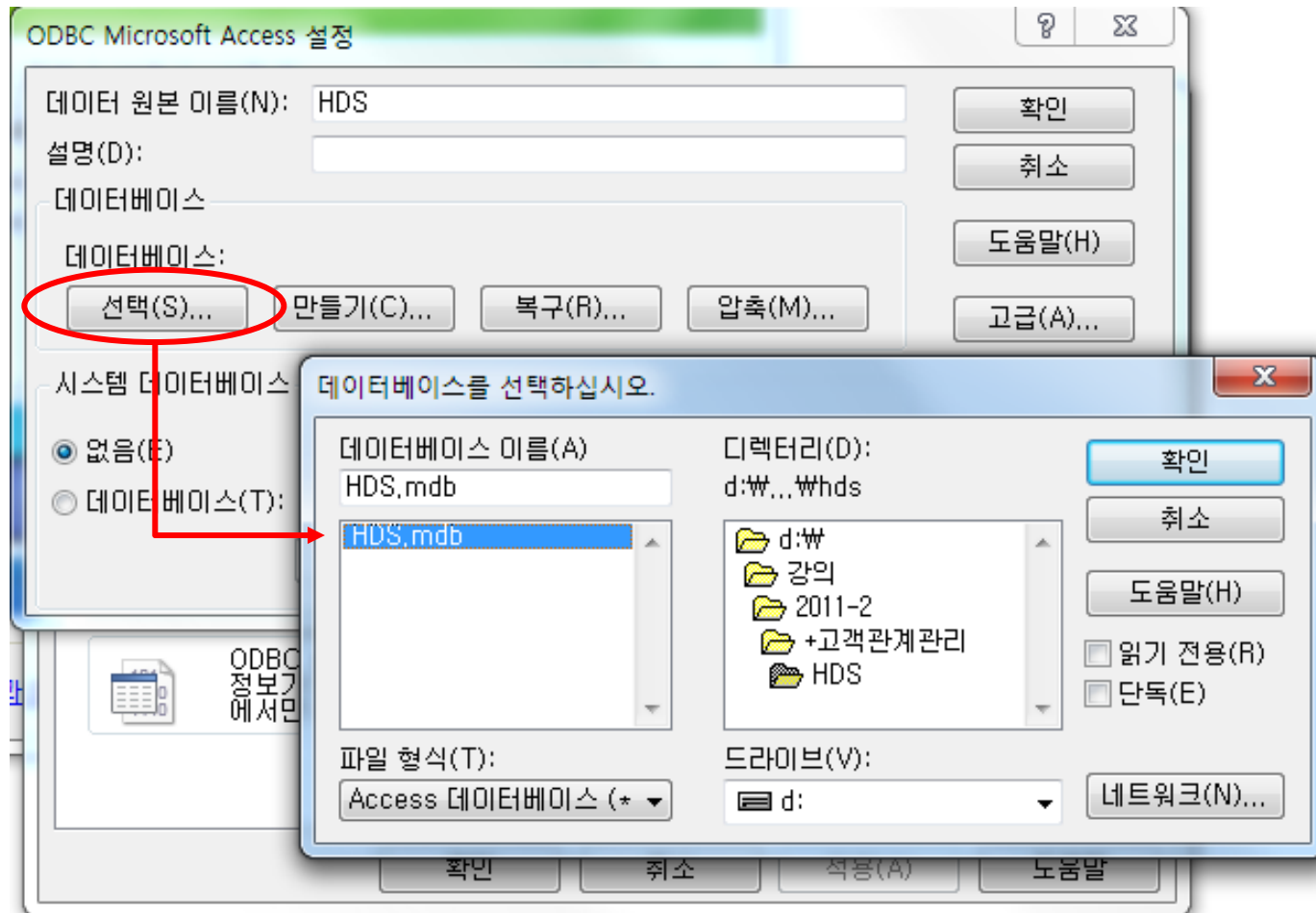
ODBC 설정 (2/4)

- 아래와 같이 "Microsoft Access Driver (*.mdb)" 선택 후 "마침" 클릭



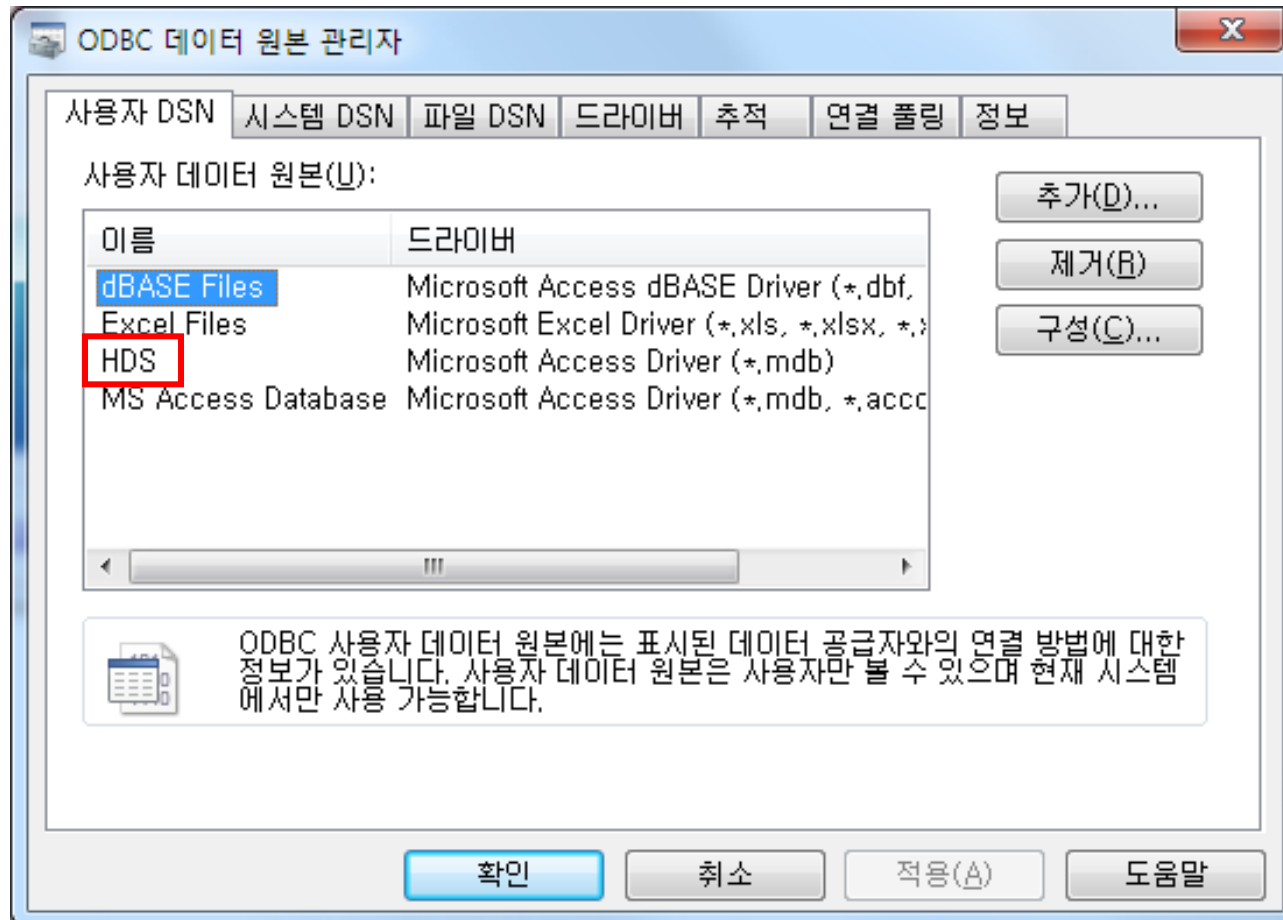
ODBC 설정 (3/4)

- 아래와 같이 데이터 원본 이름(HDS)을 입력하고 "HDS.mdb"를 선택



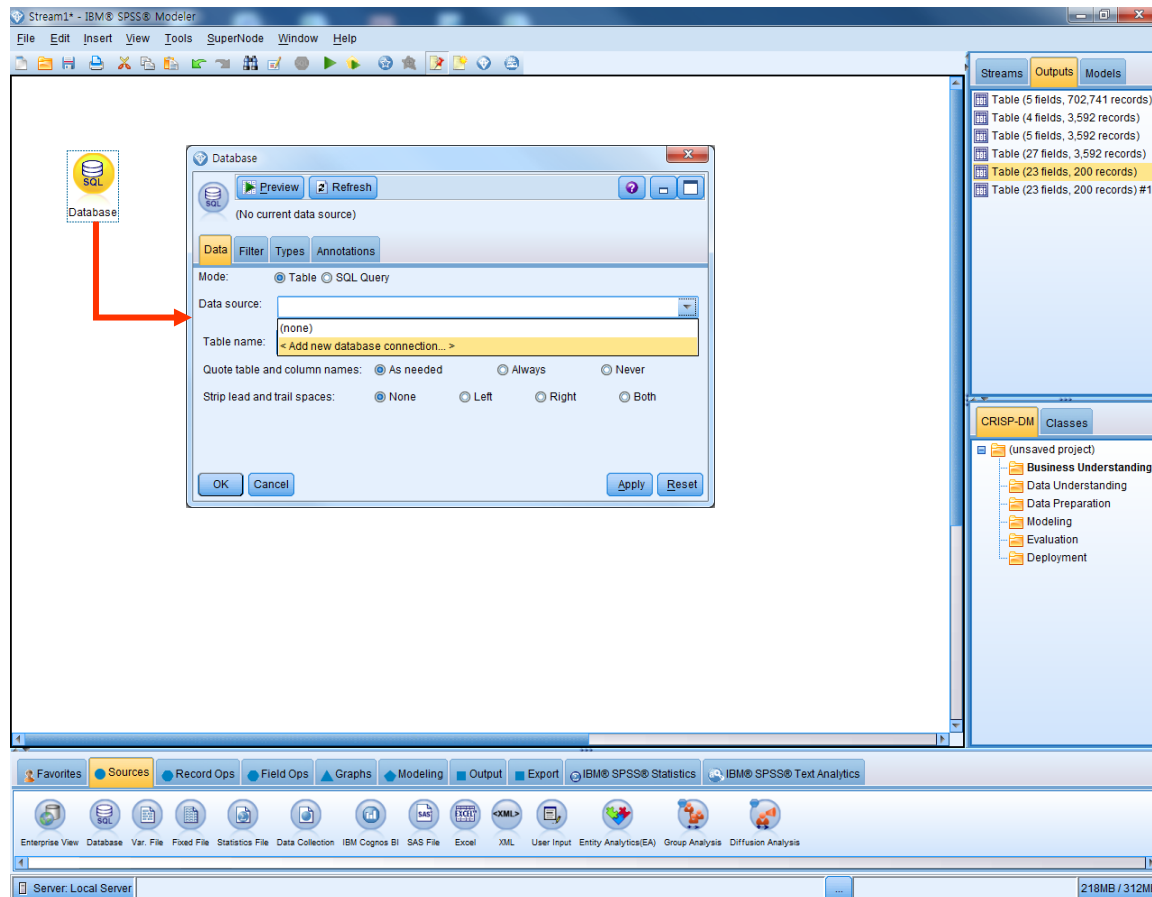
ODBC 설정 (4/4)

- 아래와 같이 HDD 이름의 ODBC가 설정된 것을 확인



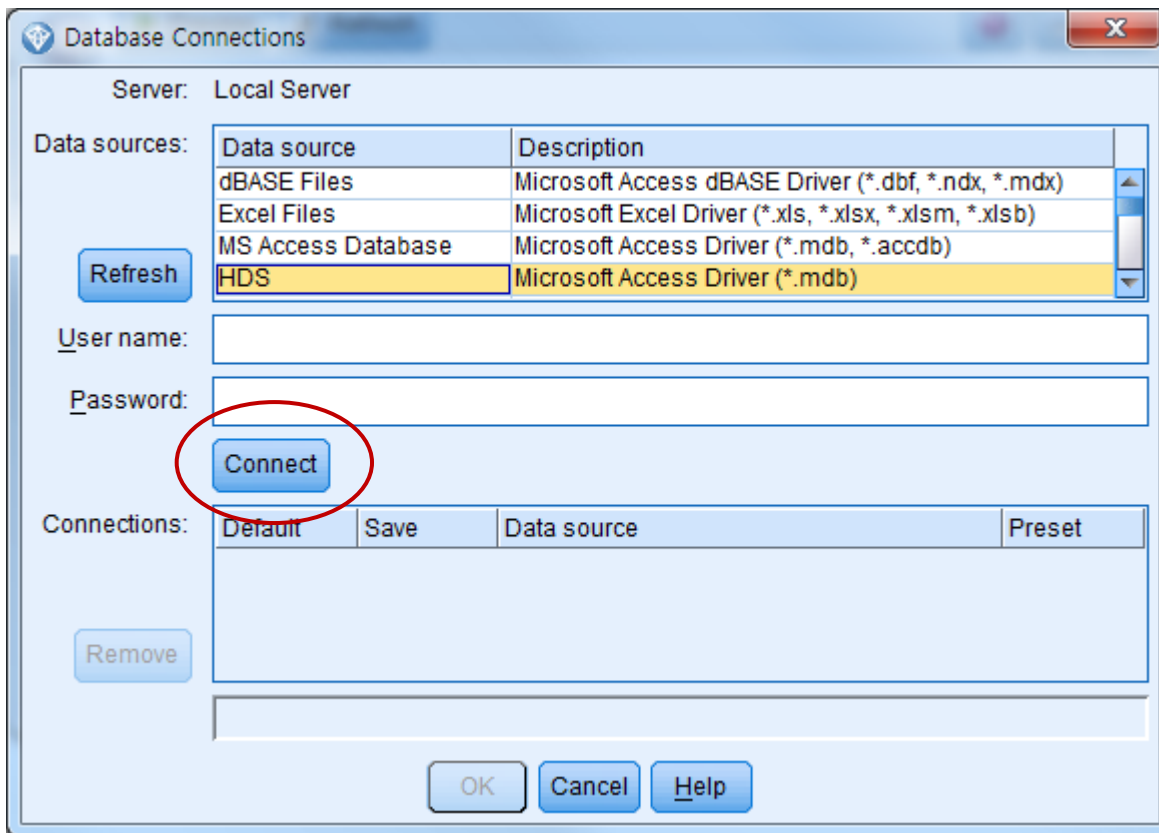
Database 노드 설정 (1/3)

- Clementine에서 소스 노드로 Database 노드를 생성한 후 아래의 화면과 같이 데이터소스를 선택



Database 노드 설정 (2/3)

- 데이터베이스 연결 화면에서 데이터소스로 (방금 전에 ODBC 설정한) HDS를 선택한 후 "연결"버튼과 "확인"버튼을 순서대로 클릭



Database 노드 설정 (3/3)

- 아래 화면과 같이 읽어오길 원하는 Table이나 Query를 선택
(이 후의 과정은 기존에 학습한 다른 Source 노드와 동일)

